ImVR: Immersive VR Teleoperation System for General Purpose

Yulin Liu[*,1], Zihao He[*,1,2], Fanbo Xiang[*,3], Runlin Guo[3], Zhiao Huang[3], Jialing Zhang[3],
Bo Ai[1], Stone Tao[1], Hao Su[1,3]
[1]University of California, San Diego, [2]Shanghai Jiao Tong University, [3]Hillbot Inc.

Abstract—**We present *ImVR*, an immersive VR-based teleoperation system that places the operator in a fully 3D virtual environment for intuitive and precise robot control. Designed for general-purpose use, *ImVR* supports a wide range of robot configurations—including single- and dual-arm setups, parallel grippers and multi-fingered dexterous hands—while operating seamlessly in both simulation and the real world. It ensures absolute alignment between the operator's hand and the robot's end-effector, offers adaptive viewpoints to enhance situational awareness and precision, and enables easy deployment across diverse robotic platforms. Experiments and user studies highlight *ImVR*'s versatility, ease of use, and effectiveness in accelerating data collection, making it a powerful tool for large-scale robot learning and real-world manipulation tasks.**

I. Introduction

The advancement of robotics increasingly relies on large-scale, high-quality datasets to train and evaluate intelligent systems. Teleoperation has emerged as a powerful solution for efficiently collecting such data, particularly for complex manipulation tasks. Recent works [14, 11] demonstrate that even a small number of human teleoperation demonstrations can seed the generation of extensive robot datasets through simulation, enabling scalable data augmentation. Furthermore, co-training with both simulated and real-world data has been shown to substantially enhance robot performance [13, 15, 22], underscoring the importance of a teleoperation system to be adapted to both simulation and real-world effortlessly.

Over the past few years, a wide range of teleoperation systems has been developed, including Virtual Reality (VR) interfaces [10, 5, 9], wearable gloves [12], and exoskeleton-based frameworks [7, 8, 21, 19]. Among these, VR-based teleoperation offers distinct advantages in terms of intuitiveness, accessibility, and potential for generalization across diverse robot platforms. VR systems enable human operators to naturally control robotic agents in real time, making them particularly suitable for collecting rich, task-relevant demonstrations in complex scenarios [10, 5, 9].

However, existing VR teleoperation systems often suffer from key limitations. Many lack precise spatial alignment between the operator's hand and the robot end-effector, leading to drift and misalignment during fine manipulation. Additionally, most systems provide static or poorly calibrated camera viewpoints, which can impair the operator's situational awareness and hinder task execution. These shortcomings limit their effectiveness for large-scale data collection and reduce applicability to tasks requiring high precision.

To address these challenges, we present *ImVR*, an advanced immersive VR teleoperation system designed to generalize effectively across various robotic platforms and real-world scenarios. *ImVR* features three key advantages:

1) Absolute alignment of the operator's hand movements with robot end-effector actions, significantly reducing spatial misalignment;
2) Adaptive viewpoint adjustments, enabling operators to intuitively inspect and manage tasks requiring precision;
3) Easy deployment in both simulation and real-world environments, with built-in tools for quickly adapting to different robot configurations with parallel grippers and dexterous hands.

Comprehensive experiments demonstrate *ImVR*'s versatility and utility, including challenging tasks in both simulation and real-world setups. User studies highlighting improved intuitiveness and effectiveness, and evaluations showing the system's ability of rapid data collection for imitation learning.

II. Method

A. System Overview

We introduce a versatile and immersive VR-based teleoperation system designed to be general-purpose, supporting multiple robots —both single- and dual-arm, parallel gripper and dexterous hand, stationary and mobile configurations (II-C)— through a unified interface for simulation and real world (II-D). At its core, the system provides an immersive experience by placing the operator in a 3D virtual world (II-B), where real-time tracking of head pose and fast high-resolution rendering (4K at 60 Hz) enables adaptive viewpoint changes and seamless user experience. This immersive environment allows precise control and intuitive manipulation, following the principle of "where your hand is, where the end-effector is," ensuring tight spatial alignment between the operator's hand and the robot's end-effector and reducing the need for compensatory motion.

Overview of the control loop is shown in Figure 2. The teleoperation modules receive the operator's motion data and translate it into robot commands. These commands are executed either in simulation or in the real world, with stereo video feedback streamed back to the operator for an immersive experience. In a simulation setup, the stereo video is rendered based on simulated environments. For real-world robot teleoperation, the stereo video is generated from point clouds captured by depth cameras.

[*] Equal contribution.

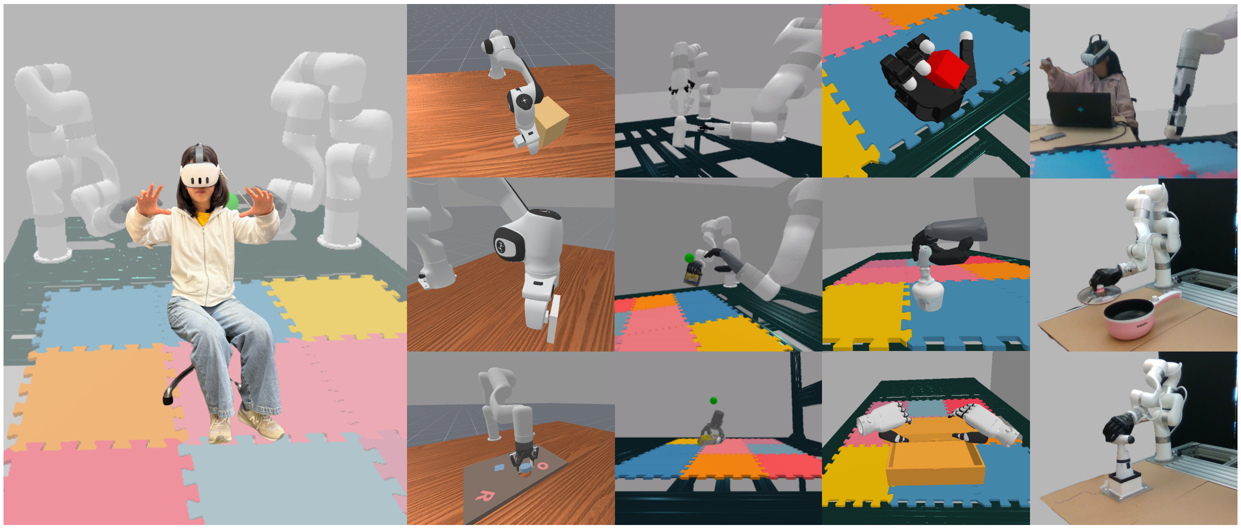

Fig. 1: One System, Two World, Multiple Robots. Our system places the operator in a fully 3D virtual world for an intuitive and immersive teleoperation experience. It is designed for general-purpose use, supporting a variety of robot setups, including single- and dual-arm configurations, grippers, three-fingered hands, dexterous hands, and both simulation and real-world environments.

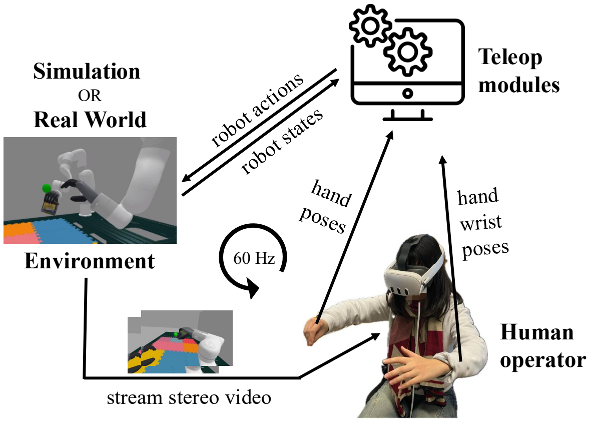

Fig. 2: Overview of our VR teleoperation system. The operator controls the robot's arm and hand through real-time tracking of wrist and hand poses, while receiving stereo video feedback. The VR devices stream human pose data to a server, which retargets and sends joint commands to the robot.

B. Placing Human in 3D Virtual World

A key feature of our system is enabling immersive teleoperation by placing human operator directly inside a fully 3D virtual world. Previous work [10, 6] relies on see-through mixed reality that overlays panels for visualization and does not provide direct access to the complete 3D scene information in simulation setups. Our method places the operator directly inside a fully rendered 3D virtual world, providing direct and complete access to spatial informatoin. This immersive design enhances intuitiveness and engagement while reducing the need for compensatory motion by operator.

We achieve this by implementing the OpenVR client protocol designed by Steam, which is compatible with all mainstream VR devices, including Meta Quest3 and Apple Vision Pro. Specifically, our system receives camera intrinsic and extrinsic parameters of the head-mounted display's stereoscopic lens, poses of controllers, and the operator's hand and wrist poses (if available), while sending stereo video streams at 4K resolution for OpenVR to display in the headset. Under the hood, OpenVR communicates with SteamVR and ALVR, which translate hardware-dependent VR implementations into the unified OpenVR client protocol. To simplify the complex SteamVR setup, we provide a Docker image for a smoother and faster setup process for users.

C. Flexible Control for Multiple Robots

We utilize a mix of control modules to provide general-purpose robot support. By modularizing control into distinct components, we ensure flexibility, extensibility and fine-grained control over diverse robots.

1) **Arm Control Module**: converts human wrist poses into robot arm joint positions. However, directly mapping the absolute orientation and position of wrist poses to robot's end-effector poses can lead to strange behaviors for a mismatch between the coordinate frames of human wrist poses and the robot's end-effector frame as illustrated in Figure 3. To address this, we provide coordinate conventors out-of-the-box for the most common robotics arms and offer a GUI and tools to assist users in quickly computing the transformation matrix for their customized robots.

Our system utilizes Closed-loop Inverse Kinematics (CLIK) algorithm, implemented with the Pinocchio library [1, 2]. To ensure smooth arm motions, we apply an SE(3) group filter to the input end-effector poses before the IK computations.

2) **Hand Control Module**: translates human finger poses into corresponding robot hand joint positions. Following [16, 4, 6], we formulate the hand motion retargeting process as an optimization problem. The objective function for this optimization is defined as follows:

$$\min_{q_t} \sum_{i=0}^{N} \|\alpha^i v_t^i - f_i(q_t)\|^2 + \beta \|q_t - q_{t-1}\|, \tag{1}$$

where q_t denotes the robot hand joint positions at time t, v_t^i is the i-th keypoint vector of human hand, and $f_i(q_t)$ gives

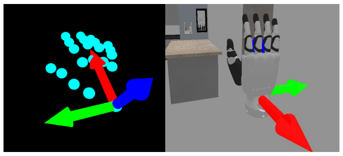

Fig. 3: Illustration of frame mismatch between human wrist poses and robot's end-effector frame. Left: human wrist frame, Right: Inspire hand frame.

the corresponding i-th keypoint vector of robot hand using forward kinematics with joint positions q_t. The scaling factor α_i compensates for the differences in hand size between the human and robot hands and treat each i-th keypoint differently as thumb finger size and pinky finger size can vary a lot, β weights the regularization term to ensure temporal consistency between consecutive joint positions. The optimization is implemented by NLopt solver [16].

For dexterous robot hands, we map vectors from human hand fingertips to palm base to corresponding vectors on the robot hand, and add extra vectors (e.g., from the thumb metacarpophalangeal joint) for improved motion accuracy. For simple grippers, we reduce to a single vector optimization between the thumb and index fingertips, enabling intuitive pinch-based gripper control. Our system includes fine-tuned configurations for common robots and calibration tools for customizing retargeting to other robots.

3) **Controller Control Module**: enables simple and effective control of gripper and wheel-based mobile robot movement. It leverages the same arm control module, but replaces input human wrist pose with VR controller's pose. By clipping the VR controller, users can intuitively trigger the closing action of the gripper for responsive grasping and release. For wheel-based mobile manipulators, we extend the module to support robot base motion by mapping button presses to forward/backward speed control and using the controller's joystick (or axis input) to control the robot's turning motion.

D. *Unified Sim-to-Real Interface*

Our system employs a unified interface for both simulation and real-world setups by aligning the robot's end-effector with the absolute position of human hand. A key challenge arises when attempting to align the human hand with the robot hand inside a VR headset, especially since the real robot may be spatially displaced in the physical world. To solve this, we project the point clouds captured by depth cameras positioned around the real robot into the VR headset. The camera poses are calibrated using EasyHec [3].

This setup ensures that both the simulation environment and the real-world point cloud are aligned in a ”digital twin” manner as illustrated in Figure 4. Though we do not require the visual textures to match the real world, critical elements such as the robot's position, forward and inverse kinematics, and control interface must be aligned. This alignment allows the same human control signals to produce identical robot actions in both the simulator and the real-world environment. Consequently, teleoperation in the real world becomes as intuitive and consistent as it is in simulation.

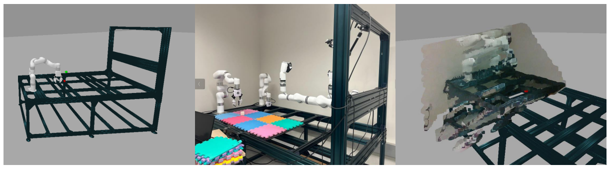

Fig. 4: Illustration of spatial alignment of simulation and real-world environments. Left: simulation environment, where robot control are tested, Middle: real-world teleoperation setup, Right: point cloud captured from the depth cameras accurately aligned with the virtual robot in the simulation, demonstrating a ”digital twin” setup.

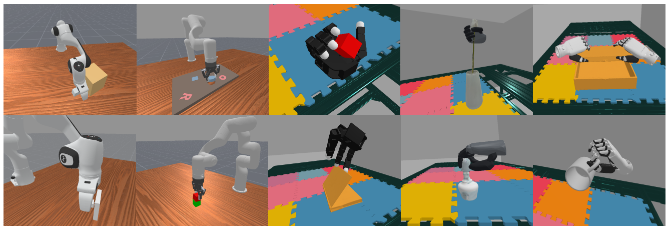

Fig. 5: Simulation Teleportation Examples. For columns left to right: (1) peg insertion and plug charger with Franka arm; (2) assemble kit and stack cube with xArm6; (3) rotate cube and open a book with Allegro hand; (4) put flower in vase and put off an alcohol burner with Ability hand; (5) open a box and pour water with the Inspire hand.

III. Experiments

In this section, we evaluate our system's versatility, data usefulness for policy learning, and the impact of two key design choices through a user study.

A. *Experiment Setup*

Our experiments span both simulation and real-world settings. In simulation, our implementation is based on the SAPIEN[20] engine and natively integrated within the ManiSkill3[18] framework, enabling seamless compatibility with a wide range of robots and tasks. For real-world teleoperation, we use three calibrated Intel RealSense cameras to capture point cloud data visual for VR rendering. One of these cameras is used for policy demonstration data. Our real-world robot platform consists of an xArm7 robotic arm equipped with an Ability Hand for dexterous manipulation tasks.

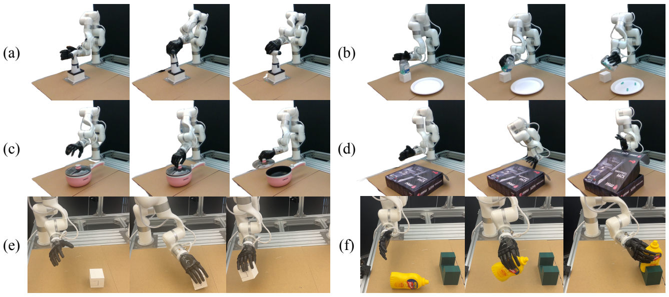

Fig. 6: Real World Teleoperation Examples with an Ability Hand mounted on an XArm7. (a) Rotating the nozzle of a watering bottle, (b) Pouring earplugs from a bottle into a tray, (c) Lifting a cooking pot lid, (d) Opening a cardboard case. (e) Pick-n-Place cube. (f) Place bottle onto a slotted rack.

Metric	Method	PickCube	PushCube	PlaceSphere	PullCubeTool	Avg.
SR (%) ↑	Keyboard	46	**100**	45	23	53.4
	Ours	**100**	**100**	**95**	**98**	**98.3**
Time (s) ↓	Keyboard	395	167	413	327	325
	Ours	**29**	**27**	**36**	**51**	**36**

TABLE I: Success rates (SR) of RFCL policies and data collection times of our system vs. keyboard control.

B. How Versatile is Our System?

Figures 5 and 6 show examples of teleoperation results in both simulation and the real world. Our system supports a wide range of embodiments, including grippers, robot hands and bimanual setups.

Notably, it handles particularly challenging tasks such as plugging in a charger or putting off an alcohol burner, demonstrating the robustness and versatility of our system. Our system provides out-of-the-box support for over 10 different robots, including various arms, robot hands. Users can teleoperate these robots directly without any additional setup. In addition, our system includes built-in visualization and calibration tools, enabling minimal-effort teleoperation of customized robots as well.

C. How Useful is our Teleoperation Data in Simulation

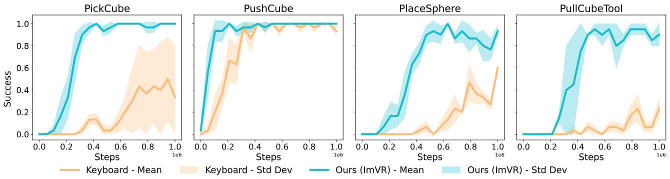

Fig. 7: Learning curve comparison of RFCL policies trained with ours vs. keyboard-collected demonstrations. The shaded area represents the standard deviation across three different seeds.

We conduct imitation learning experiments on four ManiSkill3 tabletop task, covering a diverse range of manipulation scenarios, from tool use to high-precision place sphere. To demonstrate the ability of simulation to generate scaled-up data, we employ reverse forward curriculum learning (RFCL) [17], a fast imitation learning algorthim from sparse rewards and with very few demonstrations in simulation. We collect five demonstrations using our system or the keyboard baseline.

We report the final success rate (SR) of the learned policies and total data collection time for each task, demonstrating the effectiveness of our teleoperation system in quickly collecting high-quality demonstration data for simulation-based learning. Though PushCube-v1 achieves 100 % success with either ours or keyboard-collected demonstrations, our system significantly improves sample efficiency, as illustrated in Fig. 7. When combined with RFCL, our system offers a fast and effective approach for solving tasks and generating unlimited demonstrations.

D. How Useful is Our Teleportation Data in Real World

To evaluate the real-world applicability of demonstration data collected by our *ImVR* teleoperation system, we conducted experiments on a **Pick and Place Cube** and **Place Bottle onto a Slotted Rack** tasks. We utilized the system to gather 50 trajectories for these 2 tasks. These collected trajectories then served as the training dataset for an ACT [21] imitation learning policy.

Upon training the policy, we performed an evaluation consisting of 20 trials on our physical robotic setup. The policy demonstrated a 65% success rate in completing the **Pick and Place Cube** task (Fig. 6 (e)) and 85% success rate in the **Place Bottle onto a Slotted Rack** task (Fig. 6 (f)). These result indicates that our teleoperation system is effective in generating data that can be successfully leveraged for training robotic policies for real-world applications.

E. How Intuitive is Our System for Users?

We invited five untrained operators for a user study evaluating two key design choices in our system: (i) absolute alignment of human hands with robotic end-effectors, and (ii) adaptive viewpoint change. Each operator was given 5 minutes for practice and collected 10 successful trials for three different setups on two tasks: *peg insertion using the Franka arm and Panda gripper*, and *put flower in vase using Ability dex-hand*. We report task success rate, total completion time, and average episode length. For variants without absolute alignment, the robot end-effector moved relatively to the human hand, instead of directly mirroring it. For the non-adaptive viewpoint setup, operators were instructed to keep their heads still and avoid any movement during data collection.

As shown in Fig. 8, absolute alignment reduces *peg insertion* completion time by nearly half, thanks to a more intuitive user interface and the elimination of parallax errors. Additionally, the adaptive viewpoint significantly improves success rates, as fixed viewpoints often restrict visibility and prevent close inspection of the task area. Users emphasized that adaptive viewpoint control is essential for high-precision tasks, allowing them to dynamically focus on areas requiring fine-grained manipulation—such as moving closer to specific regions for better control.

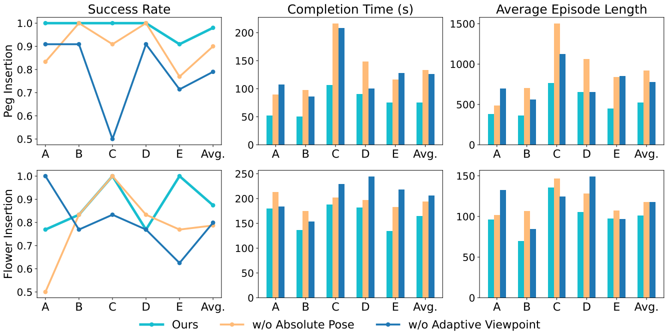

Fig. 8: User Study. It evaluates the impact of absolute position alignment and adaptive viewpoint changes on task success rates and time efficiency.

IV. Conclusion

We presented *ImVR*, a versatile and immersive VR teleoperation system that enhances the intuitiveness, precision, and adaptability. Experiments highlight its versatility, improved intuitiveness, and effectiveness for both complex manipulation and rapid data collection for imitation learning.

References

[1] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al. Pinocchio: fast forward and inverse dynamics for poly-articulated systems. https://stack-of-tasks.github.io/pinocchio, 2015–2021.

[2] 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.

[3] Linghao Chen, Yuzhe Qin, Xiaowei Zhou, and Hao Su. Easyhec: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration. *IEEE Robotics and Automation Letters*, 2023.

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

[5] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pages 2729–2749. PMLR, 2024.

[6] 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 preprint arXiv:2407.03162*, 2024.

[7] Hongjie Fang, Haoshu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. In *ICRA*, pages 15031–15038. IEEE, 2024.

[8] Hongjie Fang, Chenxi Wang, Yiming Wang, Jingjing Chen, Shangning Xia, Jun Lv, Zihao He, Xiyan Yi, Yunhan Guo, Xinyu Zhan, Lixin Yang, Weiming Wang, Cewu Lu, and Haoshu Fang. Airexo-2: Scaling up generalizable robotic imitation learning with low-cost exoskeletons. *CoRR*, abs/2503.03081, 2025.

[9] Abraham George, Alison Bartsch, and Amir Barati Farimani. Openvr: Teleoperation for manipulation. *SoftwareX*, 29:102054, 2025.

[10] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Güzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. OPEN TEACH: A versatile teleoperation system for robotic manipulation. In *CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pages 2372–2395. PMLR, 2024.

[11] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *CoRR*, abs/2410.24185, 2024.

[12] Hangxin Liu, Xu Xie, Matt Millar, Mark Edmonds, Feng Gao, Yixin Zhu, Veronica J. Santos, Brandon Rothrock, and Song-Chun Zhu. A glove-based system for studying hand-object manipulation via joint pose and force sensing. In *IROS*, pages 6617–6624. IEEE, 2017.

[13] Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, Los Angeles, CA, USA, 2025.

[14] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

[15] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[16] 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 preprint arXiv:2307.04577*, 2023.

[17] Stone Tao, Arth Shukla, Tse-kai Chan, and Hao Su. Reverse forward curriculum learning for extreme sample and demonstration efficiency in reinforcement learning. *arXiv preprint arXiv:2405.03379*, 2024.

[18] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[19] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IROS*, pages 12156–12163. IEEE, 2024.

[20] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June

2020.

[21] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

[22] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In *CoRL*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023.