IRIS: An Immersive Robot Interaction System

Author Names Omitted for Anonymous Review. Paper-ID [16]

Fig. 1: We present **IRIS**, an **I**mmersive **R**obot **I**nteraction **S**ystem designed to support various simulators and real-world scenarios.

***Abstract*—This paper introduces IRIS, an Immersive Robot Interaction System leveraging Extended Reality (XR). Existing XR-based systems enable efficient data collection but are often challenging to reproduce and reuse due to their specificity to particular robots, objects, simulators, and environments. IRIS addresses these issues by supporting immersive interaction and data collection across diverse simulators and real-world scenarios. It visualizes arbitrary rigid and deformable objects, robots from simulation, and integrates real-time sensor-generated point clouds for real-world applications. Additionally, IRIS enhances collaborative capabilities by enabling multiple users to simultaneously interact within the same virtual scene. Extensive experiments demonstrate that IRIS offers efficient and intuitive data collection in both simulated and real-world settings.**

I. Introduction

Robot learning relies on diverse and high-quality data to acquire complex behaviors [6, 54]. Recent studies indicate that models trained on more varied and complex datasets generalize more effectively across diverse scenarios [33, 46, 20]. By providing immersive perspectives and interactions, Extended Reality[1] (XR) has emerged as a promising tool for efficient and intuitive large-scale data collection in both simulation [25, 58, 42] and real-world environments [22, 13]. How-

[1]Extended Reality (XR) is an umbrella term encompassing Augmented Reality, Mixed Reality, and Virtual Reality [2].

ever, existing XR approaches face significant challenges when reused or reproduced in new scenarios, primarily due to three limitations: *asset diversity*, *platform dependency*, and *XR device compatibility*.

Current approaches [29, 25, 58, 21, 37] rely heavily on predefined sets of objects and robot models, thereby exhibiting limited *asset diversity*. Furthermore, most methods [36, 26, 42, 58] are specifically tailored to particular simulators or real-world conditions, resulting in substantial *platform dependency*. This limitation significantly reduces reusability and complicates adaptation to different simulation platforms. Additionally, existing XR frameworks [26, 39, 22, 34] are typically optimized for specific XR headset versions, leading to poor *device compatibility*. Together, these limitations severely constrain reproducibility and broader adoption of XR-based data collection and robot interaction methodologies within the research community.

To address these challenges, we propose **IRIS**—an **I**mmersive **R**obot **I**nteraction **S**ystem, demonstrated in Figure1. IRIS is a general and extensible framework that supports various simulators and real-world environments, with compatibility across different XR headsets. It is designed to generalize across six key features: *Cross-Scene*, *Cross-Embodiment*, *Cross-Simulator*, *Cross-Reality*, *Cross-Platform*, and *Cross-User*.

Cross-Scene enables XR systems to handle arbitrary simulated objects, removing constraints from predefined models. IRIS introduces a unified scene specification representing all objects as data structures with meshes, materials, and textures. This specification is transmitted to XR headsets for consistent scene rendering, with dynamic updates during simulation. Through its flexible and dynamic architecture, IRIS is also the first XR-based system that supports deformable objects manipulation. **Cross-Embodiment** is achieved by modeling robots as compositions of standard objects, enabling seamless compatibility with diverse robot embodiments without requiring specialized configurations. **Cross-Simulator** ensures compatibility with a range of simulation engines. Since the unified scene specification is simulator-agnostic, new simulators can be supported by implementing a parser to translate their scenes into this format. This flexibility is demonstrated by IRIS's support for MuJoCo [52], IsaacSim [35], CoppeliaSim [49], and Genesis [11]. **Cross-Reality** allows IRIS to operate across both simulated and real-world environments. For real-world applications, IRIS incorporates point cloud visualization using camera data, facilitating immersive data collection. **Cross-Platform** ensures compatibility across XR devices. IRIS implements its XR application using the Unity framework [51], with modular design separating visualization and interaction logic. This allows developers to deploy the system on new XR headsets by reusing visualization modules and implementing device-specific input handling. IRIS has been successfully deployed on the Meta Quest 3 and HoloLens 2. **Cross-User** supports collaborative multi-user interaction within a shared scene via a communication protocol that synchronizes XR headsets. This enables coordinated tasks and collective data collection in both virtual and real environments. Table I highlights the advantages of IRIS over existing XR-based systems across these features.

The **contributions** of IRIS are summarized as follows: (1) A unified scene specification that integrates seamlessly with multiple robot simulators, enabling consistent visualization and interaction across diverse XR headsets, while promoting reproducibility and reusability. (2) The first XR-based system to support deformable object manipulation, allowing realistic interaction and data collection for soft-body tasks. (3) A collaborative, multi-user framework for XR applications that enhances robot data collection through synchronized interactions in shared virtual or physical environments.

II. Related Work

Teleoperation-Based Data Collection on Real Robots. Collecting data using tele-operation on real robots has been explored by many previous works. Aloha [60] introduced a low-cost teleoperation system that collects real-world demonstrations for imitation learning. A bimanual workspace is set up, where leader robots are used to control the follower robots. Followup work [6] improved the performance, ergonomics, and robustness compared to the original design. In addition, a mobile version of Aloha [19] improved data collection outside of lab settings. GELLO [57] supports a variety robot arms through a 3D-printed low-cost leader robots with off-the-shelf motors. In order to tele-operate dexterous end effectors prior work has retrieved hand motion data through visual hand tracking [44] or customized gloves [54]. In contrast to IRIS, none of these approaches leverages the immersive advantages of XR.

XR-Based Data Collection in Real World. Common XR systems show virtual robots to help users understand how their movements control real robots [44]. For instance, recent work developed mobile apps to allow data collection in augmented reality without the need for XR headsets [16, 55] while XR headsets allows for more intuitive robot manipulation [58, 39, 25]. Instead of displaying the virtual robot in a third-person view, Cheng et al. [13], Iyer et al. [22] directly provide the first-person camera feed of the real robot to the user. Other systems [37, 7, 56, 10, 61, 17] visualize the real-world scene in the headset and control robot arms with controllers [61] or hand tracking [56]. XR-based data collection for dexterous hands has also been explored. For example, Arunachalam et al. [8] tracks hand motion using camera and retargets it on the real robot hand. Chen et al. [12] controls robot hand and robot arm at the same time. While these approaches do use XR, the robot data collection and interaction is limited to the real world, as no simulators used in the process.

XR-Based Data Collection in Simulation. Real robot data collection is limited by available environments and objects. Virtual data collection offers a more efficient way to gather demonstrations while providing access to extensive 3D asset libraries. For instance, DART [42] runs a cloud-based simulation, and users collect demonstrations in any virtualized environment from any location. Mosbach et al. [36]

	Cross-Scene	Cross-Embodiment	Cross-Simulator	Cross-Reality	Cross-Platform	Cross-User	Control Space
Fan et al. [17]	Limited	Single Robot	Unity	Real	Meta Quest 2	N/A	Cartesian
ARC-LfD [29]	N/A	Single Robot	N/A	Real	HoloLens	N/A	Cartesian
Zhu et al. [61]	Limited	Single Robot	N/A	Real	HTC Vive Pro	N/A	Cartesian
Jiang et al. [25]	Limited	Single Robot	N/A	Real	HoloLens 2	N/A	Joint & Cartesian
Mosbach et al. [36]	Available	Single Robot	IsaacGym	Sim	Vive	N/A	Joint & Cartesian
Holo-Dex [9]	N/A	Single Robot	N/A	Real	Meta Quest 2	N/A	Joint
ARCADE [58]	N/A	Single Robot	N/A	Real	HoloLens 2	N/A	Cartesian
DART [42]	Limited	Limited	Mujoco	Sim	Vision Pro	N/A	Cartesian
ARMADA [39]	N/A	Limited	N/A	Real	Vision Pro	N/A	Cartesian
Meng et al. [34]	Limited	Single Robot	PhysX	Sim & Real	HoloLens 2	N/A	Cartesian
Bunny-VisionPro [15]	N/A	Single Robot	N/A	Real	Vision Pro	N/A	Cartesian
IMMERTWIN [10]	N/A	Limited	N/A	Real	HTC Vive	N/A	Cartesian
Open-TeleVision [13]	N/A	Limited	N/A	Real	Meta Quest, Vision Pro	N/A	Cartesian
Szczurek et al. [50]	N/A	Limited	N/A	Real	HoloLens 2	Available	Joint & Cartesian
OPEN TEACH [22]	N/A	Available	N/A	Real	Meta Quest 3	N/A	Joint & Cartesian
Ours	Available	Available	Mujoco, CoppeliaSim, IsaacSim	Sim & Real	Meta Quest 3, HoloLens 2	Available	Joint & Cartesian

TABLE I: Comparison of XR-based system. IRIS is compared with related works in seven aspects.

collects dexterous hand manipulation data with a special glove device in physics simulations. Although Meng et al. [34] also leverages simulators, their virtual scene is a replica of the real scene, thus the flexibility of simulation is not fully exploited.

III. System Overview

This section presents the hardware and software architecture of IRIS, along with several applications explored in this paper. An overview of its paradigm is shown in Figure 2.

A. System Architecture

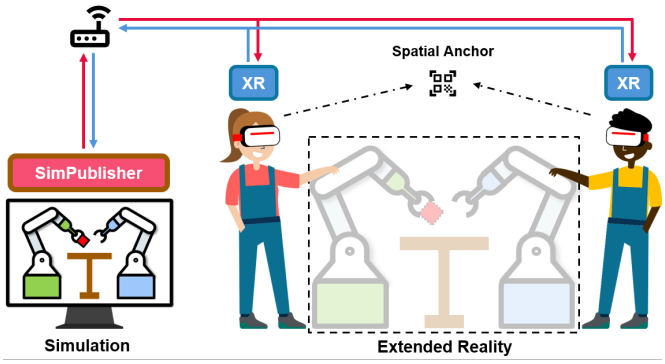
Node Communication Protocol. The IRIS system operates on simulation / sensor processing computers, XR headsets, and other programs, requiring robust network connectivity between all components. While Robot Operating System (ROS) [45] offers a general communication framework, it is not easily adaptable to Unity and XR development. Hence, IRIS built an lightweight communication protocol based on ZeroMQ (ZMQ) [5], and extended it by auto-node discovery features. To ensure node discovery, the master node broadcasts UDP messages to the broadcast port on the network at a specific frequency. The network is built via Wi-Fi or cable. When a new XR node launches, it listens to the broadcast port and receives messages from the master node. Then it extracts connection details from these messages and establishes a ZMQ connection. This protocol (Fig. 3) achieves **Cross-User** ability, ensures reliable communication, automatic reconnection, and smooth recovery from disconnections, making it ideal for dynamic multi-device XR systems.

Unified Scene Specification. To visulaize simulation scenes in XR headsets, Current solutions use predefined models in XR applications, limiting flexibility and support for new objects and robots. IRIS solves this by introducing a unified scene specification which is parsed from simulations. The unified scene specification includes all objects with their geometry, meshes, materials, and textures. All the objects is loaded in this specification using a kinematic tree structure and serialized into byte format. IRIS XR application rebuild an identical scene upon received this specification from simulation node. IRIS provides a custom Python library named *SimPublisher* that automatically generates specifications from simulation data, then it continuously collects simulation states and transmits them to headsets at a fixed frequency. This scene specification enables IRIS to all kinds of robots and objects in simulation, facilitating both **Cross-Scene** and **Cross-Embodiment** capabilities. The unified scene specification is a general definition that does not rely on any specific simulator. Hence, IRIS can be easily adapted to various simulators by implementing a new simulation parser to generate a scene specification from the simulator and a new publisher to update the states of the scene. Currently, IRIS supports scene parsers for **MuJoCo**, **IsaacSim**, **CoppeliaSim**, and **Genesis**, with the potential to be extended to other simulation engines as desired. This demonstrates that IRIS can be easily adapted to various benchmarks and simulators, highlighting its **Cross-Simulator** capability.

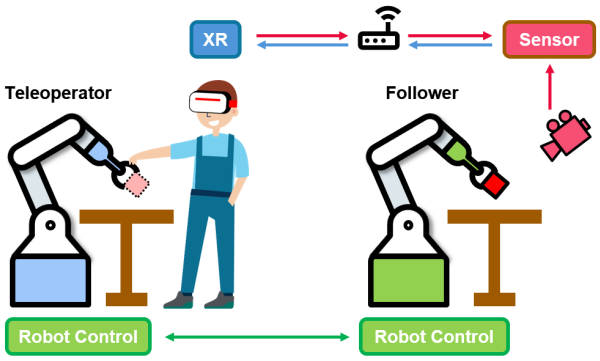
Multiple Headsets Compatibility. IRIS implements an XR application using Unity. This application can be directly deployed to other headset platforms using the Unity deployment pipeline, showcasing IRIS's **Cross-Platform** capability. Currently, IRIS has been tested on HoloLens 2 [28] and Meta Quest 3 [4]. Due to Meta Quest 3 visualization resolution is better than HoloLens 2, this paper conducted experiments and displayed XR scenes using this headset.

Intuitive Robot Control Interface. In data collection tasks or robot interaction, robot control interfaces are used to operate the robot in both simulated and real-world environments. Based on prior work [25, 22], IRIS implemented Kinesthetic Teaching (KT) and Motion Controller (MC) as its default robot controllers. These two methods were used and evaluated with other interfaces in Sec. IV-A. IRIS's flexible framework allows users to easily customize and implement additional control interfaces, including hand tracking, gloves, smartphones, and motion tracking systems. Fig. 4 shows how these two interfaces work in IRIS.

Affiliated Monitor Tools The extensibility of IRIS opens up numerous possibilities for creating new applications. IRIS includes a web-based monitoring tool for managing all XR headsets. This tool allows users to easily start and stop alignment processes, as well as rename devices. Additionally, the



(a) Interact with Robots in Simulation



(b) Interact with Robots in Real World

Fig. 2: Paradigms of the system architecture in both simulation (left) and real world (right). All the devices are connected through a Wi-Fi router. In the left image, the simulation updates the scene to all headsets using the SimPublisher. A spatial anchor is used to align the virtual scenes across different headsets. In the right image, a sensor generates a point cloud transmitted to the XR headset, allowing the operator to clearly observe the manipulated object in front of the follower robot.

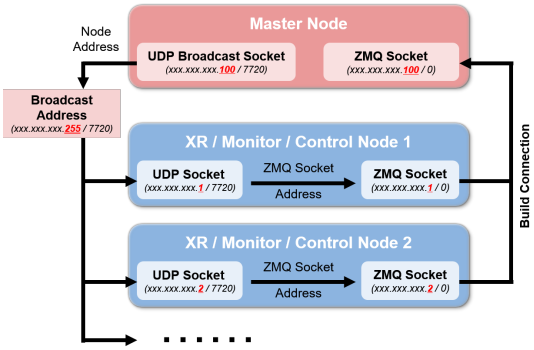


Fig. 3: The master node broadcasts UDP messages containing its details to the broadcast address (e.g., *192.168.0.255*) at port 7720. Each IP address in the diagram (e.g., *192.168.0.100/0*) represents a device's unique address on the network, where '0' indicates a dynamically assigned port provided by the operating system. XR nodes, upon startup, listen on the broadcast port (7720) to receive these messages, extract the master node's IP and ZMQ socket address, and build a stable connection. This architecture supports both request-response and publish-subscribe communication patterns, ensuring robust, multi-device connectivity with automatic reconnection capabilities.

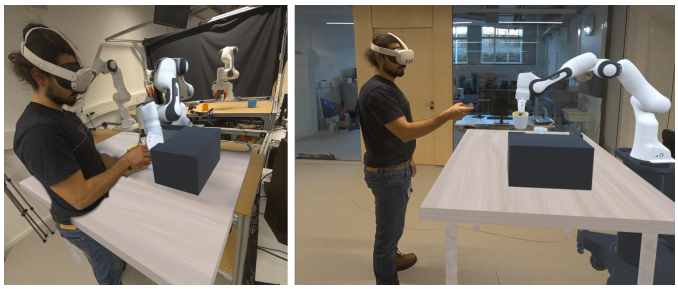


Fig. 4: This image illustrates examples of using two interfaces to control robots in simulation: Kinesthetic Teaching (left) and Motion Controller (right), shown from a third-person perspective.

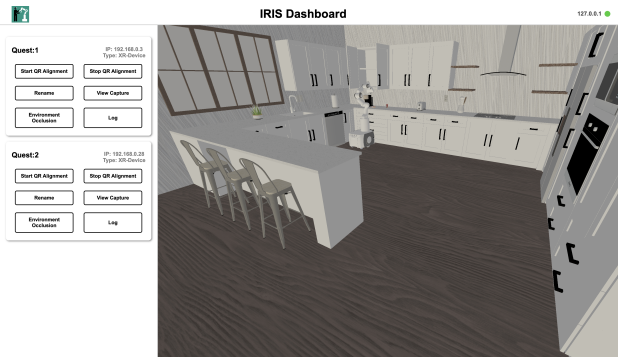


Fig. 5: The IRIS Dashboard, accessible via a web interface, allows the control and monitoring of all connected nodes. The scene streamed by IRIS is rendered on the right-hand side. The left panel displays the connected XR devices, providing an interface through which users can control all services made available by each device.

tool supports real-time scene visualization using *three.js* [1], using the unified scene specification. An example screenshot is shown in Fig. 5.

B. System Application

IRIS is a versatile platform with great potential for various applications for the robot learning and robotics research communities. This paper explores several possible applications as described below.

General Manipulation Data Collection. Through its flexible framework design, IRIS supports four simulators and various robot manipulation benchmarks. IRIS has been tested in some MuJoCo-based benchmarks including **Meta World** [59], **LIBERO** [27], **RoboCasa** [38], **robosuite** [62], **Fancy Gym** [40], and CoppeliaSim-based benchmark like **PyRep**

[23], **Colosseum** [43]. Robots can be operated using either our default controllers (Sec. III-A) or user-customized controllers.

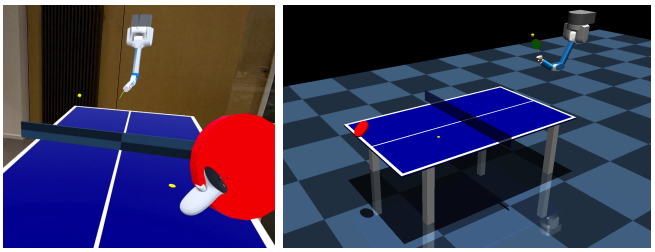
Deformable Object Manipulation. IRIS supports deformable object manipulation by dynamically updating the mesh state in real time, making it possible to train and test robotic algorithms for tasks that involve soft objects. As far as we know, no existing work has explored the manipulation of deformable objects using XR. This paper conducted a experiment to valid the data collected by IRIS based on IsaacSim. The results are in the Sec. IV-B.

Fig. 6: Collaborative manipulation in the simulation via XR. The left image shows the collaborative manipulation for hand over a hammer between two Franka Panda robots by Kinesthetic Teaching, and the right image shows that collaborative manipulation for hand over a red board between two Aloha 2 Arms by Motion Controller.

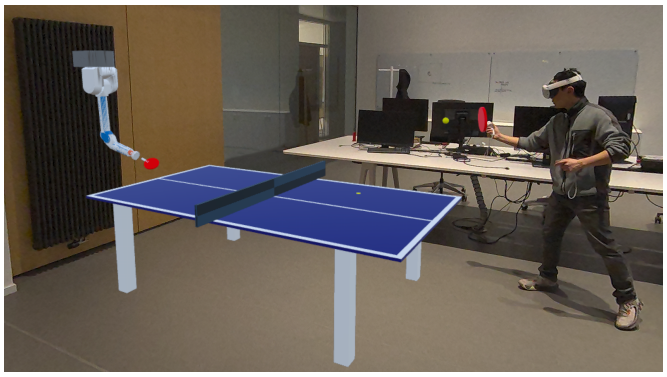
Collaborative Manipulation. Collaborative manipulation, where multiple users provide demonstrations simultaneously, is vital for human-robot systems [53]. Previous approach [44] often use multiple screens, lacking XR's immersion, and typically limit control to one person while others merely observe [50]. IRIS's communication protocol enables seamless integration of devices for controlling multiple robots in shared scenes, supporting additional XR headsets with minimal setup for dynamic collaborative environments. Fig. 6 shows collaborative manipulation for handover task.

High-Dynamic Task Data Collection and Interaction Previous works on robot tasks utilized interfaces such as keyboards [31], 3D mouse [30, 27] or smartphone [32]. However, these methods are insufficient for tasks that require complex motions. Motion controllers enable IRIS to capture user movements for complex tasks like competitive sports, where responsiveness and precision are essential. To demonstrate IRIS's capabilities, this paper designed an experiment where participants played table tennis against an episodic RL agent trained with BBRL [41] in simulation. Using a motion controller and XR headset, the participant can return the ball to the RL agent just like in the real world. Fig. 7 shows the interactive table tennis setup where a human participant play with an RL agent in a simulation. Apart from enabling human-agent interaction, this paper also leverage collected data to train policies that learn human behavior patterns for ball returns. The relevant experiment is in the Sec. IV-B.

Real World Teleoperation and Data Collection. IRIS can support both simulation and real-world applications. To minimize physical interference from humans during data collection, XR-based tele-operation is frequently utilized for data collection. Current approaches [22, 13] that utilize video streaming to XR headsets are restricted to fixed viewpoints. IRIS overcomes this limitation by projecting point cloud from depth cameras to XR headsets, ensuring both immersion and interactivity. Fig. 2b illustrates the paradigm of real-world interaction, while Fig. 8 demonstrates the practical application of IRIS in a real-world setting.



(a) Interact with RL Agent in the first-person view (left) and simulation (right)



(b) Interact with RL Agent in the third-person view

Fig. 7: Playing table tennis with RL agent in Fancy Gym environment, the RL agent policy is trained with *Deep Black-Box Reinforcement Learning* (BBRL) [41]

IV. Experiment

This section evaluates IRIS's capability to create demonstrations, focusing on the efficiency and intuitiveness of data collection pipeline. (1) How effective and intuitive is the IRIS system for data collection? (2) Can data collected by IRIS in simulation be utilized for policy training? (3) Is IRIS suitable for data collection in real-world scenarios? To answer these questions, we evaluate the performance of IRIS across three groups of tasks, including user study, policy evaluation in simulation, and real world evaluation.

A. User Study

To assess the efficiency and intuitiveness of IRIS data collection application, a user study was conducted by collecting demonstrations for LIBERO benchmark tasks [27]. Four tasks (Fig. 9 in the Appendix) were selected in the dimension of translation, rotation, and compound movement, including *close the microwave*, *turn off the stove*, *pick up the*

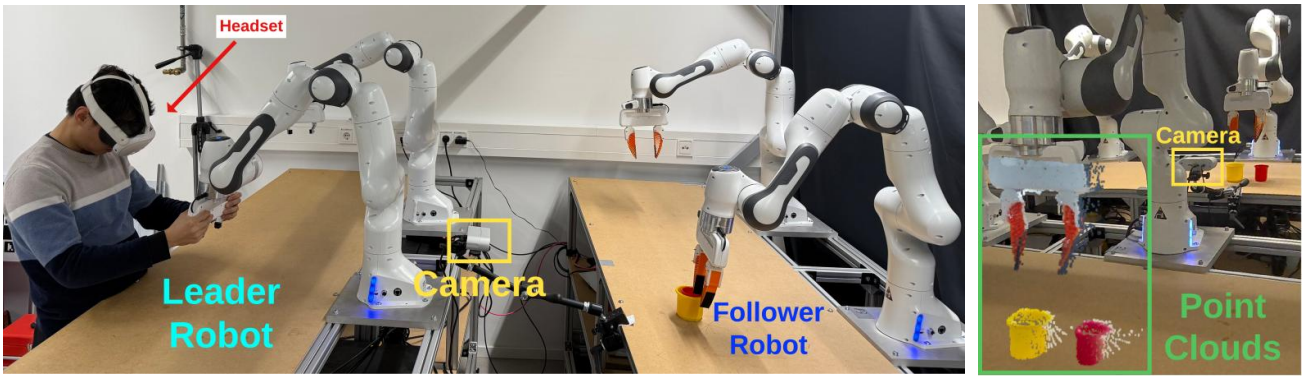


Fig. 8: Real-world application of IRIS. This setup features two Franka robots: a leader robot controlled by a user wearing a Meta Quest 3 headset and a follower robot that mirrors its movements. A depth camera captures the environment for real-time point cloud visualization in XR.

Interface	Task 1	Task 2	Task 3	Task 4
KB (LIBERO)	0.90	0.725	0.750	0.500
3M (LIBERO)	**1.00**	0.950	0.375	0.900
KT (Ours)	**1.00**	**1.00**	**0.975**	0.950
MC (Ours)	**1.00**	0.900	**0.975**	**1.00**

TABLE II: Success rate of four interfaces. KT and MC lead to higher success rate across four tasks.

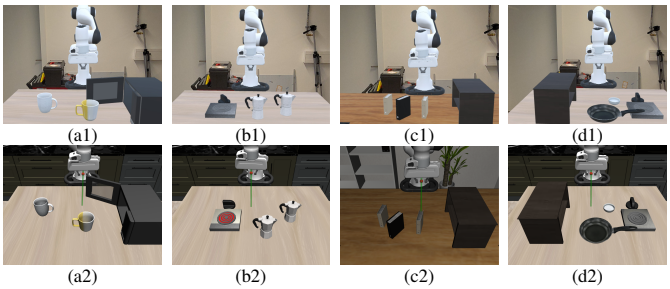


Fig. 9: Four tasks from LIBERO in simulation (top row: a1, b1, c1, d1) and corresponding view from Meta Quest 3 (bottom row: a2, b2, c2, d2): (a) Close the microwave, (b) Turn off the stove, (c) Pick up the book and place it on the shelf, (d) Turn on the stove and place the frying pan on it.

book in the middle and place it on the cabinet shelf, and *turn on the stove and put the frying pan on it*. The control interface baselines for this study are two control interfaces from LIBERO: the Keyboard (KB) and the 3D Mouse (3M), they will be compared with Kinesthetic Teaching (KT) and Motion Controller (MC) from IRIS (Sec. III-A) for the user study.

User Study Design This study involved eight participants with no prior experience using IRIS or XR headsets. They evaluated each interface through both objective and subjective metrics. Objective measurements included success rate and average time per task, while subjective assessments were gathered via a questionnaire based on the UMUX framework [18]. The questionnaire evaluates each interface by a 7-point Likert scale in four dimensions including *Experience*, *Usefulness*, *Intuitiveness*, and *Efficiency*. This paper employed the Kruskal-Wallis test [3] for a better and more robust statistical analysis for the study.

User Study Result In the result of objective metrics, Tab. II and Fig. 10 present the success rates and the average time consumed for each interface across the tasks. The result shows a success rate of over 90% across all four tasks when using the KT and MC interfaces from IRIS, and these XR-based interface significantly outperformed the non-XR interface ($p < 0.05$ [2]) in all conditions except MC in Task 2. The KT and MC interfaces consistently demonstrate lower task completion times than KB and 3M ($p < 0.05$) particularly for Task 3 and Task 4, indicating higher efficiency. The result of subjective result is shown in Fig. 11. The KT and MC interfaces consistently receive significant ($p < 0.05$) high scores than KB and 3M across all criteria, indicating positive user perception and ease of use. This study demonstrates that IRIS outperformed baseline interfaces in both objective and subjective metrics, indicating it provides a more intuitive and efficient approach for data collection.

B. Policy Evaluation in Simulation

General Manipulation To evaluate the quality of data collected using IRIS, we employ two standard imitation learning algorithms: BC-Transformer [24] and BESO [47]. These models are trained separately on datasets collected with IRIS and on the original LIBERO dataset, with results shown in Figure 12. To ensure a fair comparison, we collect the same number of trajectories using IRIS as in LIBERO, using only the MC interface (instead of KT), since LIBERO operates in Cartesian space rather than joint space. Each model is trained for 50 epochs with three random seeds to capture performance variance, and all experiments use identical training parameters. The results confirm IRIS maintains the same high data quality as the original LIBERO dataset while offering advantages in intuitive control, operational efficiency, and flexible adaptation to diverse collection scenarios.

Deformable Objects We also evaluate the data collected by IRIS from deformable object manipulation. Three tasks were

[2]level of statistical significance by Kruskal-Wallis test

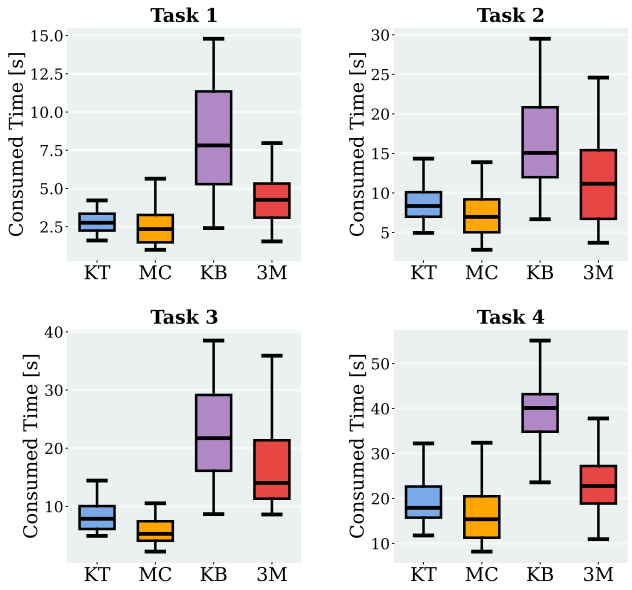


Fig. 10: This graph shows the average task completion time (in seconds) for each interface across tasks. The KT and MC interfaces consistently perform more efficiently, while the Keyboard and 3D Mouse interfaces result in longer completion times, particularly on more complex tasks.

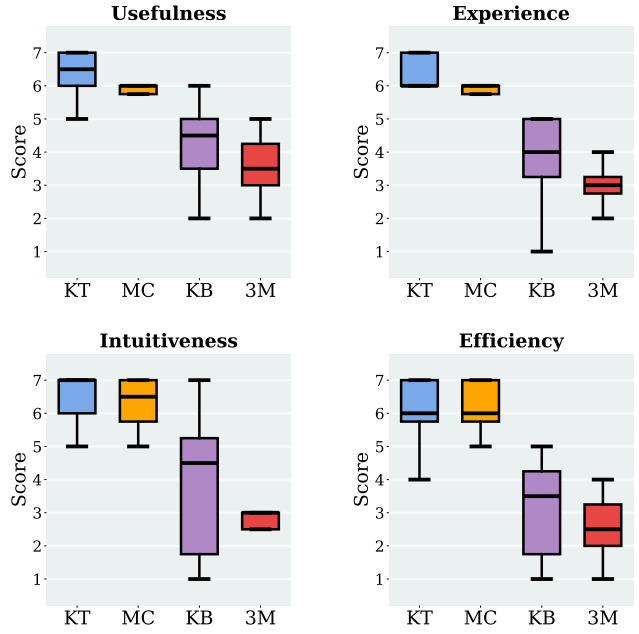


Fig. 11: Subjective evaluation scores for usefulness, experience, intuitiveness, and efficiency across four interfaces. The KT and MC interfaces perform favorably in all categories, while the Keyboard and 3D Mouse interfaces receive lower ratings, particularly in intuitiveness and efficiency.

designed to evaluate the data including *Fold Cloth*, *Lift Teddy*, and *Stow Teddy*. The policy used in this experiment is the U-Net diffusion model [14]. Observations are robot EEF pose,

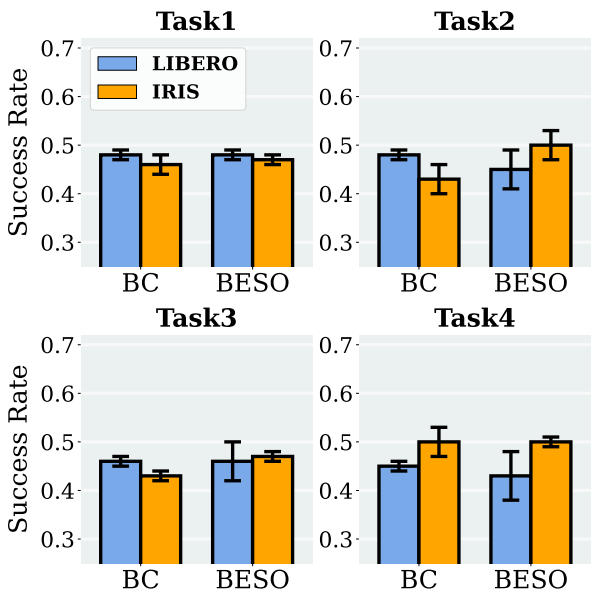


Fig. 12: Performance comparison of policies trained on different datasets across LIBERO tasks

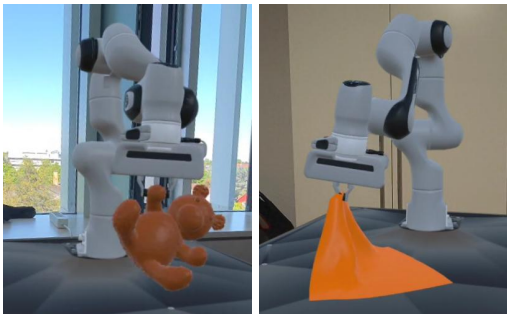
depth, and image data. The success rate of each task are *Fold Cloth*: 0.97 ± 0.018, *Lift Teddy*: 0.90 ± 0.035, and *Stow Teddy*: 0.85 ± 0.053.

Dynamic Task Data Collection To evaulate the data quality of highly dynamic task collected by IRIS, this paper uses the table tennis from Fancy_Gym [40] by motion controllers. The observation includes bat proprioceptive state and dual camera images, and the action is the desired bat position and orientation in task space. Fig. 14a shows the performance of models [48] trained on the collected data, using ball interception rate and successful return rate as evaluation metrics.

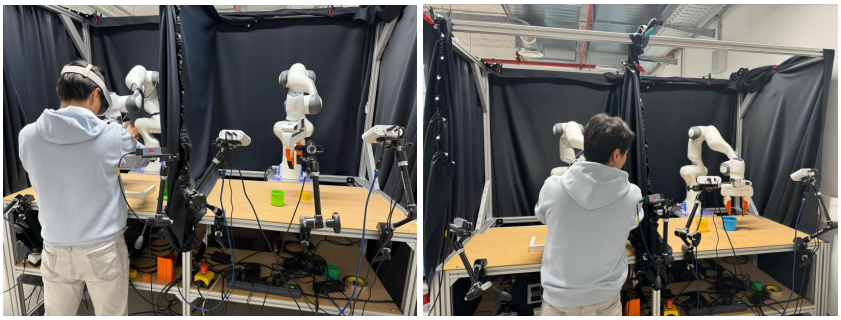
These experiments valid the data from three data collection scenarios in Section III-B, The results demonstrate that IRIS collects data of comparable quality to traditional methods, while offering significantly greater efficiency.

C. Real World Evaluation

This experiment assesses the effectiveness of IRIS for real-world data collection through two designed manipulation tasks: Cup Inserting and Picking Up Lego. IRIS was compared against Tele-Op, a widely used method for real robot data collection. For each task, 30 demonstrations were collected using both methods. These two datasets were used to train two BC-Transformer policies [24] by using the same hyper parameters. Our evaluation was twofold: data collection success rate (the percentage of successful attempts during data collection) and policy success rate after training. These reuslts (Fig. 14 b) demonstrate that IRIS provides a higher data collection control success rate, and policies trained using IRIS better quality to those trained with Tele-Op.

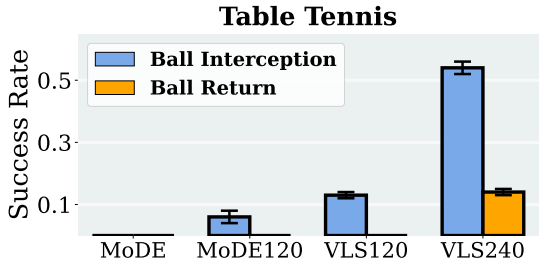


(a) Soft object manipulation in IsaacSim

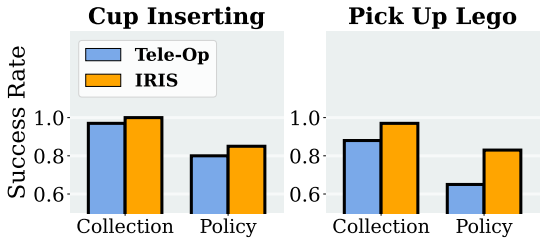


(b) Real-world data collection by IRIS(left) and Tele-Op(right)

Fig. 13: The experiment of deformable manipulation and real-world data collection



(a) Policy Performance of Table Tennis Task



(b) Policy Performance of Real World Experiment

Fig. 14: Performance evaluation of policies trained on IRIS-collected data across diverse scenarios

V. Conclusion

In this work, we introduced IRIS, an innovative framework that seamlessly integrates Extended Reality (XR) technologies with robotics data collection. IRIS addresses key challenges in reproducibility and reusability that are common in current XR-based systems. Its flexible, extendable design supports multiple simulators, benchmarks, real-world applications, and multi-user use cases. Our experiments confirm that IRIS performs effectively in three distinct data collection pipelines across simulation and real world, while validating that data collected through IRIS has comparable quality for training models. As an open-source project, IRIS codebase promotes further research and adaptation across diverse use cases and hardware platforms.

References

[1] three.js docs — threejs.org. https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene. [Accessed 30-01-2025].

[2] Extended reality - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Extended_reality, . [Accessed 11-01-2025].

[3] Kruskal–Wallis test - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_test, . [Accessed 27-04-2025].

[4] Meta Quest 3 - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Meta_Quest_3, . [Accessed 01-05-2025].

[5] ZeroMQ — zeromq.org. https://zeromq.org/. [Accessed 01-05-2025].

[6] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.

[7] Stephanie Arevalo Arboleda, Franziska Rücker, Tim Dierks, and Jens Gerken. Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–14, 2021.

[8] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[9] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[10] Florent P Audonnet, Ixchel G Ramirez-Alpizar, and Gerardo Aragon-Camarasa. Immertwin: A mixed reality framework for enhanced robotic arm teleoperation. *arXiv preprint arXiv:2409.08964*, 2024.

[11] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

[12] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human

demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024.

[13] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *8th Annual Conference on Robot Learning*.

[14] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[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 preprint arXiv:2407.03162*, 2024.

[16] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2: Training a robot without a robot. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2838–2848. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/duan23a.html.

[17] Wen Fan, Xiaoqing Guo, Enyang Feng, Jialin Lin, Yuanyi Wang, Jiaming Liang, Martin Garrad, Jonathan Rossiter, Zhengyou Zhang, Nathan Lepora, et al. Digital twin-driven mixed reality framework for immersive teleoperation with haptic rendering. *IEEE Robotics and Automation Letters*, 2023.

[18] Kraig Finstad. The usability metric for user experience. *Interacting with computers*, 22(5):323–327, 2010.

[19] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[20] Jensen Gao, Annie Xie, Ted Xiao, Chelsea Finn, and Dorsa Sadigh. Efficient data collection for robotic manipulation via compositional generalization. *arXiv preprint arXiv:2403.05110*, 2024.

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

[22] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024.

[23] Stephen James, Marc Freese, and Andrew J. Davison. Pyrep: Bringing v-rep to deep robot learning. *arXiv preprint arXiv:1906.11176*, 2019.

[24] Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, and Gerhard Neumann. Towards diverse behaviors: A benchmark for imitation learning with human demonstrations. *arXiv preprint arXiv:2402.14606*, 2024.

[25] Xinkai Jiang, Paul Mattes, Xiaogang Jia, Nicolas Schreiber, Gerhard Neumann, and Rudolf Lioutikov. A comprehensive user study on augmented reality-based data collection interfaces for robot learning. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pages 333–342, 2024.

[26] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters*, 3(1):179–186, 2017.

[27] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] lolambean. Hololens 2 hardware, March 2023. URL https://learn.microsoft.com/en-us/hololens/hololens2-hardware.

[29] Matthew B Luebbers, Connor Brooks, Carl L Mueller, Daniel Szafir, and Bradley Hayes. Arc-lfd: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3794–3800. IEEE, 2021.

[30] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024.

[31] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[32] Ajay Mandlekar, Caelan Reed Garrett, Danfei Xu, and Dieter Fox. Human-in-the-loop task and motion planning for imitation learning. In *Conference on Robot Learning*, pages 3030–3060. PMLR, 2023.

[33] Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1, 2020.

[34] Lingxiao Meng, Jiangshan Liu, Wei Chai, Jiankun Wang, and Max Q-H Meng. Virtual reality based robot teleoperation via human-scene interaction. *Procedia Computer Science*, 226:141–148, 2023.

[35] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

[36] Malte Mosbach, Kara Moraw, and Sven Behnke. Accelerating interactive human-like manipulation learning with gpu-based simulation and high-quality demonstrations.

In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 435–441. IEEE, 2022.

[37] Abdeldjallil Naceri, Dario Mazzanti, Joao Bimbo, Yonas T Tefera, Domenico Prattichizzo, Darwin G Caldwell, Leonardo S Mattos, and Nikhil Deshpande. The vicarios virtual reality interface for remote robotic teleoperation: Teleporting for intuitive tele-manipulation. *Journal of Intelligent & Robotic Systems*, 101:1–16, 2021.

[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] Nataliya Nechyporenko, Ryan Hoque, Christopher Webb, Mouli Sivapurapu, and Jian Zhang. Armada: Augmented reality for robot manipulation and robot-free data acquisition. *arXiv preprint arXiv:2412.10631*, 2024.

[40] Fabian Otto, Onur Celik, Dominik Roth, and Hongyi Zhou. Fancy gym. URL https://github.com/ALRhub/fancy_gym.

[41] Fabian Otto, Onur Celik, Hongyi Zhou, Hanna Ziesche, Vien Anh Ngo, and Gerhard Neumann. Deep black-box reinforcement learning with movement primitives. In *Conference on Robot Learning*, pages 1244–1265. PMLR, 2023.

[42] Younghyo Park, Jagdeep Singh Bhatia, Lars Ankile, and Pulkit Agrawal. Dexhub and dart: Towards internet scale robot data collection. *arXiv preprint arXiv:2411.02214*, 2024.

[43] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

[44] 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*, abs/2307.04577, 2023. URL https://api.semanticscholar.org/CorpusID:259367735.

[45] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[47] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[48] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *arXiv preprint arXiv:2407.05996*, 2024.

[49] E. Rohmer, S. P. N. Singh, and M. Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[50] Krzysztof Adam Szczurek, Raul Marin Prades, Eloise Matheson, Jose Rodriguez-Nogueira, and Mario Di Castro. Multimodal multi-user mixed reality human–robot interface for remote operations in hazardous environments. *IEEE Access*, 11:17305–17333, 2023.

[51] Unity Technologies. Unity - Manual: Unity 6 User Manual — docs.unity3d.com. https://docs.unity3d.com/6000.0/Documentation/Manual/UnityManual.html. [Accessed 31-01-2025].

[52] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[53] Albert Tung, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Learning multi-arm manipulation through collaborative teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9212–9219. IEEE, 2021.

[54] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[55] Jun Wang, Chun-Cheng Chang, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Eve: Enabling anyone to train robots using augmented reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706288. doi: 10.1145/3654777.3676413. URL https://doi.org/10.1145/3654777.3676413.

[56] Xingchao Wang, Shuqi Guo, Zijian Xu, Zheyuan Zhang, Zhenglong Sun, and Yangsheng Xu. A robotic teleoperation system enhanced by augmented reality for natural human–robot interaction. *Cyborg and Bionic Systems*, 5: 0098, 2024.

[57] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. *arXiv preprint arXiv:2309.13037*, 2023.

[58] Yue Yang, Bryce Ikeda, Gedas Bertasius, and Daniel Szafir. Arcade: Scalable demonstration collection and generation via augmented reality for imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2855–2861. IEEE, 2024.

[59] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-

world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

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

[61] Yaonan Zhu, Bingheng Jiang, Qibin Chen, Tadayoshi Aoyama, and Yasuhisa Hasegawa. A shared control framework for enhanced grasping performance in teleoperation. *IEEE Access*, 2023.

[62] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.