ARCap: Collecting High-quality Human Demonstrations for Robot Learning with Augmented Reality Feedback

Sirui Chen*, Chen Wang*, Kaden Nguyen, Li Fei-Fei, C. Karen Liu

***Abstract*— Recent progress in imitation learning from human demonstrations has shown promising results in teaching robots manipulation skills. To further scale up training datasets, recent works start to use portable data collection devices without the need for physical robot hardware. However, due to the absence of on-robot feedback during data collection, the data quality depends heavily on user expertise, and many devices are limited to specific robot embodiments. We propose ARCap, a portable data collection system that provides visual feedback through augmented reality (AR) and haptic warnings to guide users in collecting high-quality demonstrations. Through extensive user studies, we show that ARCap enables novice users to collect robot-executable data that matches robot kinematics and avoids collisions with the scenes. With data collected from ARCap, robots can perform challenging tasks, such as manipulation in cluttered environments and long-horizon cross-embodiment manipulation. ARCap is fully open-source and easy to calibrate; all components are built from off-the-shelf products. More details and results can be found on our website: stanford-tml.github.io/ARCap**

I. Introduction

Developing robots to assist with domestic tasks has the potential to enhance human quality of life and augment human capabilities. To achieve this, robots must be able to manipulate everyday objects in unstructured and often cluttered environments. Imitation learning using human demonstrations has made significant progress in recent years. Demonstration data collected via teleoperated robotic systems provide precise, in-domain observation and action pairs, enabling effective robot policy learning through supervised learning [45]. However, the requirement for a robotic system and a skilled human operator upfront significantly limits the accessibility and scalability of data collection.

Alternatively, human demonstrations can be collected using portable systems without the need for physical robot hardware [6, 35, 38]. These systems leverage human dexterity and adaptability to directly manipulate objects in-the-wild, facilitating the creation of large-scale, diverse human demonstration datasets. However, due to the absence of robot hardware, whether the collected demonstrations are useful for training robot policies is not immediately apparent without going through a multi-step process. First, the differences in embodiment between humans and robots require data retargeting. Second, the retargeted data must be validated by replaying the motion on the actual robot interacting with real objects. Finally, the robot policy must be trained using validated data. The success of demonstrations critically depends on the demonstrator's experience and awareness of the disparities between the robot and human geometry and capabilities. Failures can occur at the retargeting stage due to the robot's joint and speed limitations, during the validation stage due to incidental collisions, or at the policy training stage due to the mixture of invalid data.

(a). Human Data Collection (b). AR Feedbacks (c). Robot Rollout

Fig. 1: **ARCap System Overview.** (a) Collect human hand motion data. (b) Provide real-time AR feedback, visualizing a virtual robot retargeted to the human hand in AR display. (c) Rollout robot policies trained with the collected data.

This leads us to ask: Is there a way to inform users of potential failures during data collection so they can adjust and collect higher-quality data? One key observation from on-robot teleoperation is that when humans see incorrect robot motions, they quickly adjust the way of teleoperation to correct the error. This strong *visual feedback* helps users collect data that is executable and suitable for the robot's embodiment. Given the success of visual feedback in teleoperation, the question arises: Can we *simulate* similar feedback in portable data collection systems to guide users in collecting high-quality demonstration data?

We propose ARCap, a novel data collection system that retargets and visualizes a robot's motion in real-time, providing the demonstrator with instant visual feedback during data collection and guiding them to collect robot-executable demonstrations. This is achieved by leveraging augmented reality (AR) technology, both as an interactive display and a powerful sensor that captures the user's view of the environment. Using the AR display, we can simulate the robot's kinematics, overlay it in the headset, and provide visual cues for potential failure modes, such as exceeding the robot's joint or speed limits (e.g., the virtual robot fails to follow the human hand). Additionally, with recent advances in scene reconstruction in AR devices, we can perform collision checking between the virtual robot and the reconstructed environment. When a collision is detected, the system warns users with a blinking effect in the display and haptic vibration, prompting them to adjust their movements and leave enough space for the robot's embodiment. Besides

* Equal contribution
Stanford University, Department of Computer Science

improving data quality, ARCap can simulate any robot embodiment, enabling data collection for different robots (e.g., parallel-jaw grippers, multifinger dexterous hands). In our user study, we found that ARCap enables novice users without any prior data collection experience to collect high-quality data. These data are sufficient to train imitation learning policies, even for tasks like manipulation in cluttered environments—tasks that were impossible with previous systems lacking feedback. We also demonstrate that ARCap can collect data across different robot embodiments, enabling robots to accomplish challenging long-horizon manipulation tasks, such as stacking multilevel Lego towers.

II. Related work

Learning from Demonstrations. Imitation Learning (IL) has proven effective in enabling robots to perform various manipulation tasks [1, 2, 4, 11, 13, 20, 24, 33]. While traditional IL methods like DMP and ProMP [23, 29, 30, 34] are highly sample-efficient, they face challenges in handling high-dimensional observation spaces. In contrast, recent IL approaches leveraging deep neural networks can learn policies directly from raw image inputs [14, 27, 49], even for complex robotic systems with bimanual manipulators [17, 42, 46]. Although these methods are effective, scaling the amount of training data remains a significant hurdle. Teleoperation, a commonly used method for data collection in recent studies [3, 5, 9, 14–16, 18, 19, 21, 25, 26, 31, 32, 39, 41, 44, 45, 48]. Many low-cost teleoperation systems built upon VR controller or hand tracking[5, 9, 18, 22] and master-slave joint mapping[12, 15, 37, 42, 45, 47] were widely used. However, despite the low-cost nature of these action input devices, collecting data using teleoperation still requires the presence of a actual robot, which makes them expensive to distribute on a large scale. In contrast, our approach follows the recent fashion of collecting robot data without robot hardware [7, 10, 35, 38, 40], allowing us to scale up the training data more efficiently.

Data Collection System without Robots. Collecting data in the wild without the presence of a robot and training robots with that data has become an attractive direction to lower the total cost of the system. Prior works such as [7, 35, 38] proposed low-cost, in-the-wild data collection systems. Compared to directly using human video for training[36], these systems capture more fine-grained human movement and have helped robots to achieve complex tasks such as tea preparation[38], plate wiping[7, 38] and using air fryer[35]. Our ARCap system is another portable, in-the-wild data collection system; compared to existing systems, it provides visual, haptic feedback, which helps users without any data collection experience be aware of the embodiment gap between robots and humans. The most related work to ARCap is AR2-D2 [10, 40]. ARCap, however, focuses on providing *real-time* visual feedback and onboard collision checking using the reconstructed scene map. Additionaly, ARCap helps users collect data for different robot embodiments such as parallel-jaw grippers and multi-finger dexterous hands, by visualizing the retargeted robot in the AR display.

III. Method

ARCap is an AR-based data collection interface and policy learning framework designed to transfer human hand motion capture data to robot control policies. The main features of ARCap's system design are:

- **Real-time feedback.** AR provides real-time visualization of the robot states, guiding users to collect high-quality and robot-reproducible demonstration data without physical robots.
- **Cross-embodiment.** AR visualization supports both parallel-jaw grippers and multifinger dexterous hands, allowing users to collect data for different types of robot hardware using the same system.
- **Portability.** With a self-contained power supply, storage, and wireless tracking, the system enables data collection in-the-wild.

In this section, we first describe the system design that enables these features, followed by the training policies for controlling real robots.

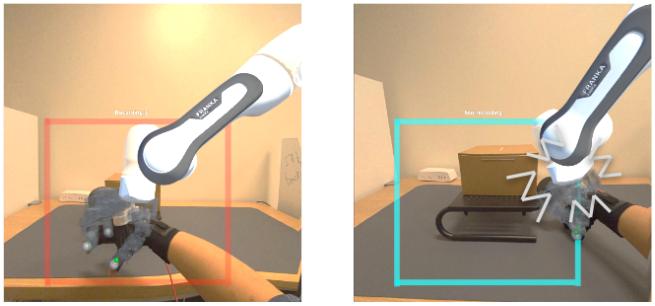
A. ARCap System Design

Recent advancements in portable robot data collection interfaces [7, 35, 38] have made it possible to scale up robot data collection without needing a physical robot. However, since there is no real-time feedback from a robot during the data collection process, there is no guarantee that the collected data will be reproducible on an actual robot. Several failure modes have been observed: (1) Humans move too quickly for the robot to replicate; (2) Size differences between humans and robots cause the robot to collide with the environment, even when humans do not; (3) One data collection system is designed for one robot embodiment, requiring redesigns for different robot end-effectors. These observations begs the question: How can we alert humans about these issues during data acquisition and guide them to collect robot-reproducible data?

Informative AR Feedbacks. In ARCap, we implement both visual and haptic feedback to inform users about camera visibility, robot kinematics, joint speed limits, and potential collisions between the robot and the environment.

a) Real-time visibility checking: One common failure mode for imitation learning is that the scene of manipulation is not always visible. This issue occurs frequently because RGB-D cameras used by robots usually have a narrower field of view compared to the cameras used for data collection—in our case, the passthrough cameras in Quest 3. To help the demonstrator always keep the manipulation scene within the field of view of the depth camera during data collection, we render a rectangular frame to visualize the actual field of view of the RGB-D camera, as shown in Fig.2. When collecting data, users needs to actively keep the scene inside the frame to ensure visual data is being recorded properly.

b) Real-time retargeting: When collecting data for a particular robot, the robot may have significantly different kinematics compared to the human arm and hand. To remind users about the kinematic limit, we rendered a virtual robot



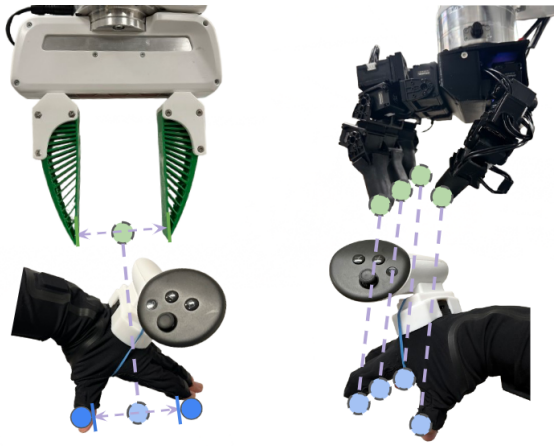
(a). Recording (b). Collision Warning

(c). Fast Motion Warning (d). Out of frame, out of sight

Fig. 2: **Visualization of AR Feedback.** (a) Normal data recording: the red frame indicates visible region of the RGB-D camera. (b) Collision warning: when the virtual robot collides with the environment, the controller on the human gloves vibrates, and the frame blinks blue. (c) Fast motion warning: when the user moves faster than the robot's speed limits, the frame turns yellow. (d) Users can check if target objects are within camera's view during data collection.

in AR and retargeted it to the user's hand. Different end-effectors may have different retargeting methods, as we will discuss in the next section. Before data collection, the user will place the virtual robot at a fixed location in the world frame. During data collection, the end-effector of the virtual robot will track the user's hand; whenever the user uses their hand to interact with an object in the scene, they need to consider whether the virtual robot could perform such an action. For example, for a virtual robot equipped with a parallel jaw gripper, if the user tries to reorient an object using finger gaiting, the action performed by the virtual robot will appear to be invalid, as shown in the attached video. As each joint of the robot arm has its speed limit, the virtual robot also implements such limits and won't exceed the speed limit to track the user's input. If the user moves their hand too fast, there will be a significant visual mismatch between the user's hand and the robot end-effector; the rectangular frame will also blink yellow to remind the user the robot has its speed limit.

c) Real-time collision checking: To remind users about the potential collisions between the robot and the environment, we also check the collision between the actual scene and the virtual robot. We found it is hard for humans to perceive depth accurately with passthrough cameras; only watching the movement of the virtual robot is not enough to avoid collision. We add extra haptic collision feedback when the virtual robot collides with the pre-scanned static scene



(a). Parallel-jaw Gripper (b). Dexterous Hand

Fig. 3: **Cross-Embodiment Data Collection.** (a) ARCap can collect data for parallel-jaw grippers by guiding the user to form their hands into a gripper-like shape. If the user changes the hand gesture, the retargeting error will be large. (b) For a multi-finger dexterous hand, ARCap retargets the robot's fingertips to match the human fingertips, with the robot's wrist orientation determined by the orientation of the controller mounted on the user's gloves.

by vibrating the mounted controller. The rectangular frame will also flash to provide a stronger collision feedback signal as shown in Fig.2.

Using these real-time feedback signals, user can adjust their data collection strategies or delete the demonstration with severe constraint violations.

Cross-Embodiement with One System. ARCap can visualize various end-effectors retargeted to the user's hand, enabling the collection of data for different robot embodiments without requiring hardware modifications. For any new robot embodiment, ARCap can be used for data collection as long as a retargeting process is in place that allows the robot to repeat human demonstrations. We present two real-time retargeting processes for different end-effectors attached to the Franka Panda arm: (1) Leap Hand, a fully actuated, four-finger dexterous hand, and (2) the Fin-ray gripper, a compliant parallel jaw gripper.

- **Dexterous hand.** Similar to [38], we match the fingertips of a dexterous hand to the fingertips of a human in the world frame using inverse kinematics. The inverse kinematics problem is solved in two steps. It first solves the leap hand wrist pose to match the human wrist pose provided by the quest controller and then solves the robot fingertip positions to match human fingertip positions tracked by the Rokoko data glove. As each finger of the leap hand has one redundant degree of freedom, we need to add null space regulation to encourage a natural hand posture and avoid self-collision between fingers. We use null space IK solver from Pybullet[8], which solves the current joint angle based on the previous solution in real-time.
- **Parallel jaw gripper.** For the parallel jaw gripper, the

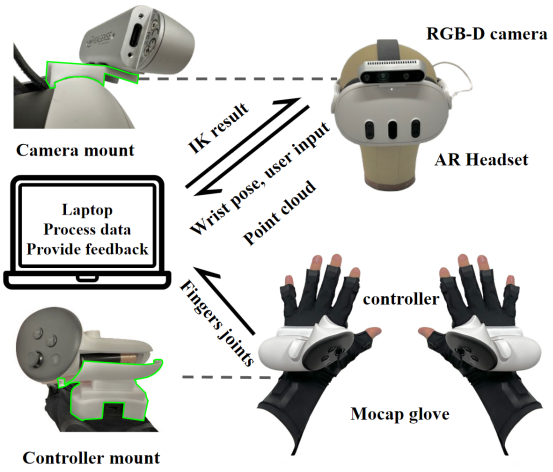


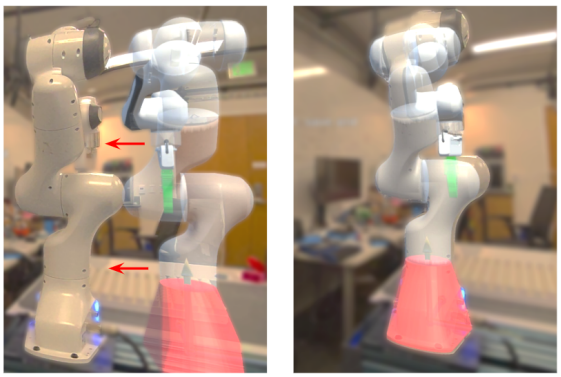
Fig. 4: **ARCap System Layout.** The user wears an AR headset and motion capture gloves, with controllers mounted on the gloves for tracking the 6D pose of the palms. Data is stored on a laptop carried in the backpack.

user mimics it by using their index finger and thumb. As shown in Fig.3, the midpoint of the gripper tips is aligned with the midpoint between the user's index finger and thumb, while the wrist orientation tracked by the controller sets the gripper's orientation. Since the gripper can only fully open or close, its state is determined by the distance between the user's index finger and thumb. If the distance is greater than gripper width at open state, it is set to open; otherwise, it is set to close. On real robots, the gripper responds to open and close commands at 1Hz. In our retargeting process, if the user opens and closes their hand too frequently, the virtual gripper won't open or close till 1s from changing to the previous state.

Portable and Reproducible Design. ARCap is designed to be a low-cost, portable system that is easy to reproduce and calibrate, while accurately capturing detailed hand motions. It also ensures user comfort during various tasks with minimal obstruction. To achieve these goals, ARCap is built around the Meta Quest 3 VR headset, as shown in Fig.4. The headset serves as both a display for feedback and a sensor hub, providing spatial tracking for itself and both controllers. A RealSense D435 camera is mounted on top of the headset using a 3D-printed bracket to capture 3D visual information, which is stored as point clouds. Since accessing the internal Quest 3 camera is difficult, future versions of ARCap could leverage the built-in RGB-D camera of an AR headset.

For wrist and hand motion capture, Quest 3 controllers are attached to the top of Rokoko data gloves. The controllers track wrist position and orientation relative to the headset, while the data gloves capture fingertip positions relative to the wrist. Using the headset's built-in SLAM function, we can access both visual and motion data within a world frame.

Calibrating the system can be time-consuming due to the need to fine-tune relative transformations between components. To streamline this process, the camera is mounted directly to the headset, and the controllers are attached to the gloves with unique 3D-printed mounts, allowing future



(a). Before Calibration (b). After Calibration

Fig. 5: **AR-based Camera Calibration.** When calibrating the camera, users align the virtual robot's base with the actual robot's base.

ARCap setups to reuse the same calibration parameters. A laptop is connected to process visual data and provide additional storage for collected data. Like DexCap [38], ARCap system is portable and can be carried in a backpack, enabling data collection without external infrastructure.

B. Imitation learning

1) Data processing: ARCap records the following data:

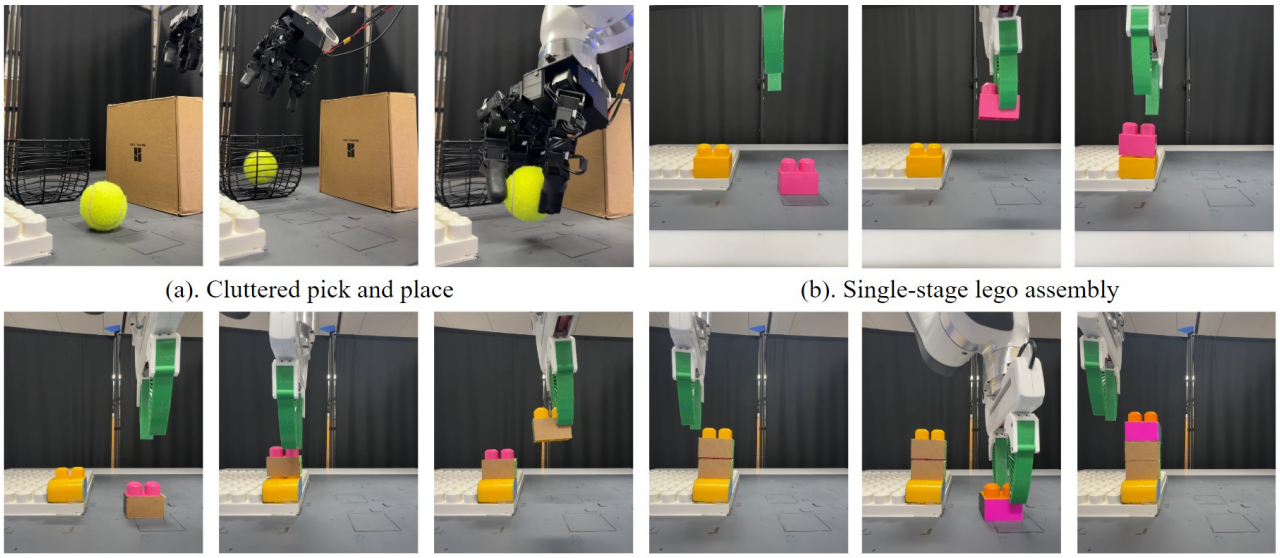
- Colored point cloud in the camera frame
- Joint angle for the virtual robot solved by IK
- Headset pose in the world frame
- Virtual robot pose in the world frame

The collected data can be used for imitation learning with a simple post-processing procedure. We first transform every data into the world frame. For point clouds, we further crop them in the world frame to remove background objects and the desktop. In the collected data, the hand and arm of a human user are visible. To reduce the visual gap, we superimpose a point cloud of the virtual robot visible by the depth camera in our point cloud dataset. After processing, all data for a single task will be stored in one hdf5 file.

2) Training and testing: With processed data, we use diffusion policy for imitation learning. For encoding 3D point cloud, similar to [38, 43], a simple point net is used to compress colored point clouds into a latent vector. After that, the latent vector is concatenated with the current joint angle of the robot arm and hand as observation $\boldsymbol{o}$. The generated action $\boldsymbol{a}$ consists of the target joint angles of both robot arm and hand; for dex hand, $\boldsymbol{a}$ consists of the target joint angles of each finger; for parallel jaw gripper, $\boldsymbol{a}$ include a binary open and close command. Our training and testing pipeline is built upon robomimic[28], a unified framework for robot imitation learning. When testing trained policy, we can utilize the ARCap system to simplify the hand-eye calibration process. As shown in Fig.5, to compute the camera pose related to the robot base, we align the base of the virtual robot to the base of the actual robot in the ARCap application.

IV. Experiments

We design experiments to answer the following questions:



(a). Cluttered pick and place

(b). Single-stage lego assembly

(c). Long-horizon lego assembly

Fig. 6: **Experiment Tasks.** (a) Pick-and-place in a cluttered scene with a dexterous robot hand. (b) Single-stage Lego assembly with a parallel-jaw gripper. (c) Long-horizon assembly of three Lego blocks.

Q1 Does ARCap enable general users to collect higher-quality data

Q2 Can data collected by ARCap help robots to manipulate under a cluttered environment?

Q3 Can data collected by ARCap work on the robots with significantly different embodiments?

Q4 Does data from ARCap good enough for achieving long-horizon manipulation?

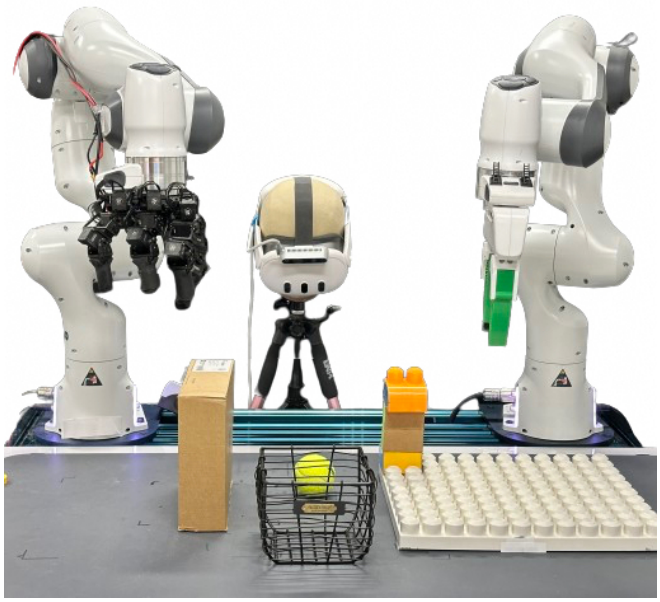


Fig. 7: **Setups for Real Robot Evaluation.** During evaluation, we mounted the headset on a tripod and connected its camera to the robot workstation. The trained policy takes point cloud observations from the camera attached to the headset and outputs actions to control the robots.

A. Experiment setup

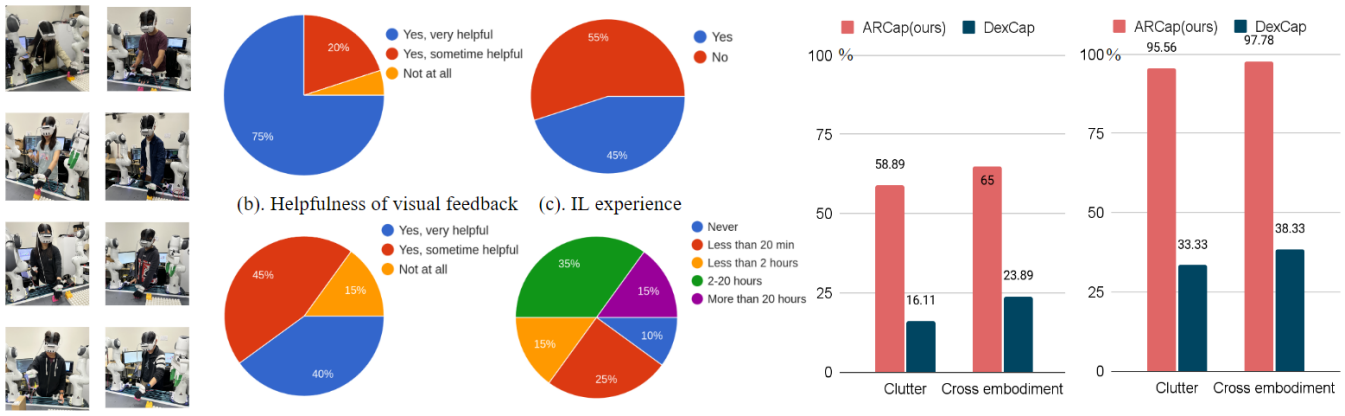
We used two Franka Panda arms in our experiment, one attached with a Leap hand and another one attached with a Fin-ray gripper. Two robots shared the same workspace. For data collection, a Quest 3 headset runs a Unity application for visualization and data streaming, and a Windows laptop with an i5-13200H CPU is used for solving IK and storing data. For training and testing autonomous policy, we use a workstation with a single RTX3090 GPU and i7-13700 CPU. When testing, we calibrate the camera using the above-mentioned process and put the headset on a dummy head to serve as an RGB-D camera, as shown in Fig.7.

B. User study

To answer the **Q1**, we conduct a user study and invite 20 participants to use our new system ARCap with visual haptic feedback and the previous system DexCap, which has no feedback. Users have different exposure to VR/AR devices, and half of them have no data collection or robot learning experience; Fig.8.(c,e) shows the demographic of our participants. Moreover, none of the participants used either ARCap or DexCap before participating in this study.

Test participants are asked to collect data for two tasks as shown in Fig.6: (1). Picking and placing a tennis ball with obstacles using a dexterous leap hand. (2). Assembling a single Lego block with a fin-ray parallel jaw gripper. The first task aims to test whether feedback from ARCap can help the user avoid collision under a cluttered environment; the second task aims to test whether feedback can help the user collect valid data under different end-effector embodiment. Each task has 3 initial states, and the subject needs to collect 3 trajectories on each initial state.

From experience with training imitation learning policies, trajectory reproducibility and scene visibility are two essential factors determining the quality of collected demonstra-



(a). User study snapshots (b). Helpfulness of visual feedback (c). IL experience (d). Helpfulness of haptic feedback (e). VR/AR experience (f). Trajectory replay success rate (g). Scene visible rate

Fig. 8: **Results of User Study.** (a) We invited 20 users with different backgrounds to collect data with DexCap [38] and ARCap. (b)-(e) Survey results of user experience. (f)-(g) On-robot evaluation results with collected data.

tions. In our user study, to quantitatively measure the quality of collected data, we test whether robots can replay the collected trajectory and accomplish the same task, as well as whether the manipulation scene is always visible during the data collection process.

Fig.8 shows the replay success rate and scene visible rate of both tasks; ARCap achieve over 40% higher replay success rate and over 60% higher scene visible rate compared to DexCap. In our evaluation, we found that ARCap frequently prevents failures caused by collisions or kinematic limits in both tasks. It can also significantly reduce failure cases caused by users ignoring the gripper closing speed limit. A post-survey was conducted by the participants and summarized in Fig.8.(b,d). Results show that most participants found visual and haptic feedback useful in helping them improve data collection strategies.

C. Manipulation in cluttered environment

To verify whether data collected by ARCap can actually help robot imitation learning to achieve manipulation in a cluttered environment. We collected two 30-minute datasets using both systems and trained two diffusion policies on each of them. These two datasets are collected by the authors of this paper, who are familiar enough with both ARCap and DexCap. After training, we evaluate the policy using 20 trials with different initialization. Shown in Tab.I, ARCap can achieve a 35% higher success rate compared with DexCap, and no collision ever happens when testing ARCap policy. We also merge 30-minute data crowd-sourced from multiple first-time users during user study and train an autonomous policy from them. ARCap policy can achieve a 60% success rate across 3 designated initial states, while DexCap policy failed everytime across different trials, shown in Tab.I

Cluttered pick and place	Expert		User	
DexCap [38]	0.25		0	
ARCap	**0.7**		**0.6**	
Long-horizon lego assembly	Stage 1	Stage 2	Stage 3	Full
DexCap [38]	0.5	0.15	0.05	0.0
ARCap	**0.7**	**0.8**	**0.85**	**0.4**

TABLE I: Success Rate of Autonomous Policy

D. Long horizon manipulation with different embodiments

To answer **Q3** and **Q4**, we also show that ARCap can collect high-quality data with embodiment significantly differ than human and help robots achieve the task using imitation learning. We demonstrate this capability using a long-horizon, three-stage Lego assembly tasks with parallel jaw gripper as demonstrated in Fig.6.(c). This task is challenging as it requires policy to learn different grasp and assembly actions for Lego blocks at different stages. We used both DexCap and ARCap to collect two datasets of one-hour human manipulation and trained two policies based on them. We first evaluate the success rate independently at different stages. In this evaluation, humans reset the Lego tower to one stage prior to assembly after each trial. As shown in Tab.I, ARCap can achieve 70%, 80%, and 85% success rates at stages 1, 2, and 3. We also evaluate the policy on assembling all 3 stages autonomously in which humans always reset the Lego tower to the base level; ARCap policy achieves a 40% success rate in full autonomous assembly, which is, on average, 51% higher than the DexCap policy. The policy could also react at different stages when humans disassemble the Lego tower, as shown in our supplementary video.

V. Conclusions and Future Work

We propose ARCap, a portable data collection system that allows users without prior experience to collect high quality data across different embodiments via visual, haptic feedback. Using ARCap, we can teach robot manipulation in cluttered environments and achieve horizon cross embodiment manipulation with imitation learning. In the future, with additional design in feedback and retargeting process, ARCap also record human torso movement to collect data for mobile robots or humanoids. Currently, user improves their data collection strategies passively from feedback; with VLM, ARCap could also provide instruction for users to actively improve their data collection strategies and efficiency.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*, Springer, 2008, pp. 1371–1394.

[3] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[4] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[5] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[6] C. Chi *et al.*, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.

[7] C. Chi *et al.*, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.

[8] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, 2015, p. 1.

[9] R. Ding *et al.*, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," *arXiv preprint arXiv:2407.03162*, 2024.

[10] J. Duan, Y. R. Wang, M. Shridhar, D. Fox, and R. Krishna, "Ar2-d2: Training a robot without a robot," *arXiv preprint arXiv:2306.13818*, 2023.

[11] P. Englert and M. Toussaint, "Learning manipulation skills from a single demonstration," *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 137–154, 2018.

[12] H. Fang *et al.*, "Low-cost exoskeletons for learning whole-arm manipulation in the wild," *arXiv preprint arXiv:2309.14975*, 2023.

[13] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on robot learning*, PMLR, 2017, pp. 357–368.

[14] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 492–499, 2019.

[15] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[16] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh, "Efficient data collection for robotic manipulation via compositional generalization," *arXiv preprint arXiv:2403.05110*, 2024.

[17] J. Grannen, Y. Wu, B. Vu, and D. Sadigh, "Stabilize to act: Learning to coordinate for bimanual manipulation," in *Conference on Robot Learning*, PMLR, 2023, pp. 563–576.

[18] J. van Haastregt, M. C. Welle, Y. Zhang, and D. Kragic, "Puppeteer your robot: Augmented reality leader-follower teleoperation," *arXiv preprint arXiv:2407.11741*, 2024.

[19] T. He *et al.*, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[20] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, 1398–1403 vol.2.

[21] L. Ke, A. Kamat, J. Wang, T. Bhattacharjee, C. Mavrogiannis, and S. S. Srinivasa, "Telemanipulation with chopsticks: Analyzing human factors in user demonstrations," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 11 539–11 546.

[22] A. Khazatsky *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[23] J. Kober and J. Peters, "Learning motor primitives for robotics," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 2112–2118.

[24] J. Kober and J. Peters, "Imitation and reinforcement learning," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 55–62, 2010.

[25] T. Lin *et al.*, "Learning visuotactile skills with two multifingered hands," *arXiv:2404.16823*, 2024.

[26] A. Mandlekar *et al.*, "Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1048–1055.

[27] A. Mandlekar *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," in *5th Annual Conference on Robot Learning*, 2021.

[28] A. Mandlekar *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[29] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[30] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26, Curran Associates, Inc., 2013.

[31] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor poli-

cies via consistency distillation," *arXiv preprint arXiv:2405.07503*, 2024.

[32] Y. Qin *et al.*, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," *arXiv preprint arXiv:2307.04577*, 2023.

[33] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[34] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*, Springer, 2006, pp. 261–280.

[35] N. M. M. Shafiullah *et al.*, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[36] K. Shaw, S. Bahl, and D. Pathak, "Videodex: Learning dexterity from internet videos," *CoRL*, 2022.

[37] K. Shaw *et al.*, "Bimanual dexterity for complex tasks," in *8th Annual Conference on Robot Learning*, 2024.

[38] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv preprint arXiv:2403.07788*, 2024.

[39] C. Wang, R. Wang, A. Mandlekar, L. Fei-Fei, S. Savarese, and D. Xu, "Generalization through hand-eye coordination: An action space for learning spatially-invariant visuomotor control," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 8913–8920.

[40] J. Wang, C.-C. Chang, J. Duan, D. Fox, and R. Krishna, "Eve: Enabling anyone to train robot using augmented reality," *arXiv preprint arXiv:2404.06089*, 2024.

[41] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," *arXiv preprint arXiv:2309.13037*, 2023.

[42] S. Yang *et al.*, "Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation," *CoRL*, 2024.

[43] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv preprint arXiv:2403.03954*, 2024.

[44] T. Zhang *et al.*, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 5628–5635.

[45] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[46] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[47] T. Z. Zhao *et al.*, "Aloha unleashed: A simple recipe for robot dexterity," in *8th Annual Conference on Robot Learning*.

[48] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *arXiv preprint arXiv:2210.11339*, 2022.

[49] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Conference on Robot Learning*, PMLR, 2023, pp. 1199–1210.