AirExo-2: Scaling up Generalizable Robotic Imitation Learning with Low-Cost Exoskeletons

Hongjie Fang[*,1], **Chenxi Wang**[*,2], **Yiming Wang**[*,1], **Jingjing Chen**[*,1], **Shangning Xia**[1,2], **Jun Lv**[1,2], **Zihao He**[1], **Xiyan Yi**[1], **Yunhan Guo**[1], **Xinyu Zhan**[1], **Lixin Yang**[1], **Weiming Wang**[1], **Cewu Lu**[1,2,3,†], **Hao-Shu Fang**[1,†]
[1]Shanghai Jiao Tong University [2]Noematrix [3]Shanghai Innovation Institute
[*]Equal Contribution [†]Corresponding Authors

http://airexo.tech/airexo2

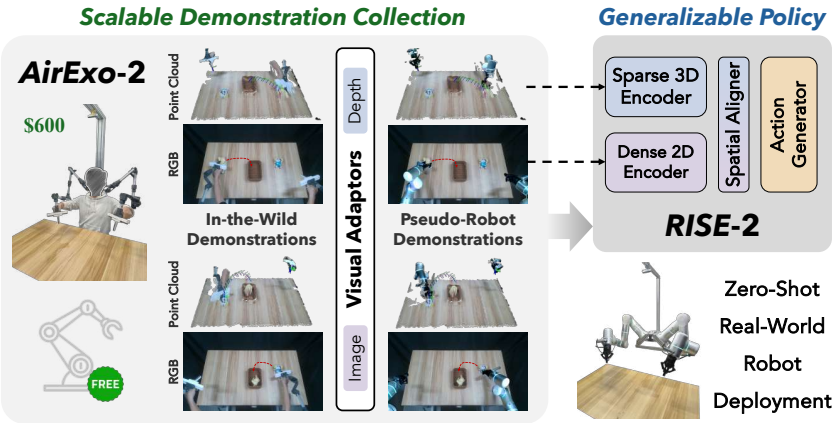


Figure 1: Overview of the *AirExo*-2 System and the *RISE*-2 Policy. ***AirExo*-2** enables scalable in-the-wild demonstration collection and adaptation. Visual adaptors convert these into pseudo-robot demonstrations for policy training. The generalizable ***RISE*-2** policy learns from this data alone and achieves zero-shot deployment on real dual-arm robots, matching the performance of policies trained on teleoperated robot data.

Abstract: Scaling up robotic imitation learning for real-world applications requires efficient and scalable demonstration collection methods. While teleoperation is effective, it depends on costly and inflexible robot platforms. In-the-wild demonstrations offer a promising alternative, but existing collection devices have key limitations: handheld setups offer limited observational coverage, and whole-body systems often require fine-tuning with robot data due to domain gaps. To address these challenges, we present ***AirExo*-2**, a low-cost exoskeleton system for large-scale in-the-wild data collection, along with several adaptors that transform collected data into pseudo-robot demonstrations suitable for policy learning. We further introduce ***RISE*-2**, a generalizable imitation learning policy that fuses 3D spatial and 2D semantic perception for robust manipulations. Experiments show that ***RISE*-2** outperforms prior state-of-the-art methods on both in-domain and generalization evaluations. Trained solely on adapted in-the-wild data produced by ***AirExo*-2**, the ***RISE*-2** policy achieves comparable performance to the policy trained with teleoperated data, highlighting the effectiveness and potential of ***AirExo*-2** for scalable and generalizable imitation learning.

Keywords: Robotic Manipulation, Scalable Data Collection, Generalizable Imitation Policy

[*] galaxies@sjtu.edu.cn, cxwang85@gmail.com, sommerfeld@sjtu.edu.cn, jjchen20@sjtu.edu.cn
[†] lucewu@sjtu.edu.cn, fhaoshu@gmail.com

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Scaling up generalizable robotic imitation learning in the real world is key to developing robust policies for practical applications [4, 7]. While teleoperation is widely used for collecting demonstrations, it requires physical robots to record both observations and actions for *robot-centric* demonstrations, making it costly and difficult to scale. To address this, recent work explores alternative, low-cost data sources such as human videos [5, 57, 91, 97, 99, 108, 112] and in-the-wild demonstrations [19, 29, 88, 100, 115]. These *human-centric* demonstrations eliminate the need for a robot during collection, reducing costs and improving scalability. Human video approaches typically infer actions via hand or object pose estimation [36, 76, 97, 113], while in-the-wild demonstrations use devices that interface human motion with robot-relevant data during task execution.

Current in-the-wild demonstration devices fall into two main categories: handheld [19, 24, 87, 88, 115] and whole-body devices [17, 29, 53, 100]. Handheld devices are typically designed with end-effectors identical to those of the robot. By equipping both the handheld devices and the robot with in-hand cameras, these methods leverage in-hand observations to ensure visual consistency. However, they face two key issues: (1) reliance on visual SLAM for pose estimation leads to action inaccuracies (§4.3); and (2) limited camera field of view hinders depth perception during interactions (Appendix C.3). On the contrary, whole-body devices offer more accurate action capture and flexible observations, but they typically require fine-tuning with teleoperated data [29, 100], as in-the-wild demonstrations still exhibit a domain gap compared to real robot data.

To address these challenges, we present the ***AirExo*-2** system for large-scale in-the-wild demonstration collection and adaptation. As illustrated in Fig. 1 (left), we introduce demonstration adaptors that align both observations and actions from in-the-wild demonstrations with the robot domain, enabling direct use in imitation learning. On the hardware side, we build upon AirExo [29] with a redesigned exoskeleton featuring a stronger mechanical structure and improved calibration, enhancing action precision and system robustness. We conduct comprehensive analyses to evaluate the system's efficiency and accuracy in demonstration collection.

Beyond scaling up in-the-wild demonstration collection, developing a robust policy is key to fully leveraging these demonstrations. We introduce ***RISE*-2**, a generalizable policy that fuses 2D and 3D perception, as illustrated in Fig. 1 (right). ***RISE*-2** outperforms prior imitation policies in both in-domain and generalization settings. Notably, it matches the performance of teleoperated-data-trained policies using only in-the-wild demonstrations collected and adapted by ***AirExo*-2** — without any robot data. Under equivalent data collection time, it even surpasses teleoperated baselines. These results highlight the strong potential of combining ***AirExo*-2** and ***RISE*-2** for scalable, generalizable imitation learning. The hardware and codebase are open-sourced at the project website.

2 Related Works

Scaling up Demonstration Collection. Demonstration data is fundamental to advancing generalizable imitation learning in robotic manipulation. Recent studies on data scaling laws [59] reveal that imitation learning in robotics follows similar scaling trends observed in natural language processing [49] and computer vision [35, 77] fields, highlighting the importance of scaling up demonstration collection. Teleoperation is the most direct and widely used approach for collecting real-world robot demonstrations, as it captures data in the robot's native observation domain and action space. However, scaling up teleoperated demonstration collection is challenging, primarily limited by the high cost and infrastructure demands of deploying more robots. Consequently, constructing large-scale teleoperated robotic datasets [9, 10, 21, 27, 45, 52, 98, 110] requires substantial human labor and physical resources. Moreover, teleoperation is often less intuitive than natural human manipulation, resulting in higher user learning costs [65] and lower demonstration quality [13].

To overcome these limitations, recent research explores generating [43, 72, 105, 107] or augmenting [1, 40, 48, 70, 103] demonstrations automatically in simulation, which facilitates demonstration collection but often necessitates sim-to-real adaptation for real-world deployment. Other approaches leverage internet-scale human videos for representation learning [2, 67, 68, 69, 73, 81, 93, 118] or policy pre-training [5, 12, 32, 80, 109, 114], yet accurately converting 2D human videos into robot demonstrations with 3D trajectories and robot states remains a challenge [71]. Alternatively, in-the-wild data collection uses specialized human-operated devices — such as hand-held grippers [19, 24,

87, 88, 115], cameras [23, 104], VR/AR systems [17, 50, 53], motion-capture gloves [100], and exoskeletons [29, 53] — to translate human motions into robot actions.

Learning from In-the-Wild Demonstrations. Despite promising in scalability, the kinematic gap and the visual gap pose obstacles in learning from in-the-wild demonstrations [29, 53]. **(1)** The kinematic gap refers to discrepancies in motion translation between humans and robots, which can lead to inaccuracies in action execution. This gap can be addressed through visual or mechanical methods. Visual approaches typically use off-the-shelf pose estimation frameworks from commercial devices [17, 19, 23, 24, 50, 87, 88, 100, 104] or structure-from-motion [86, 115] to extract robot poses from demonstrations. In contrast, mechanical approaches [29, 53] employ kinematically isomorphic devices that capture robot poses through angle encoder readings. **(2)** The visual gap, on the other hand, arises from the presence of specialized devices and human hands in the captured visual data, whereas robot demonstrations must accurately reflect the robot itself. Hand-held devices [19, 24, 87, 88, 115] mitigate this by using in-hand cameras and the same end-effector during deployment to maintain visual consistency. Other methods address the visual gap by incorporating robot demonstrations for fine-tuning or co-training [23, 29, 50, 104], or by employing human-in-the-loop strategies to collect corrective behaviors during policy deployment [17, 100].

Generalizable Manipulation Policy. Direct learning from in-the-wild demonstrations highlights the need for a generalizable manipulation policy capable of transferring learned skills to the robot during real-world deployment. A generalizable policy can adapt to new domains or environments, even with limited training data from restricted domains [111], and should generalize across variations in camera perspectives, backgrounds, objects, and embodiments. While many behavior-cloning-based robotic manipulation policies [18, 30, 31, 78, 90, 123] perform well in-domain, they often fail in out-of-distribution scenarios, leading to compounded errors and task failures [102, 111]. Large-scale pre-training on real-world robot demonstration data can improve generalization [6, 9, 21, 45, 55, 62, 74, 106, 127], but it does not fully address the inherent limit of the policy in adapting across diverse domains. Two promising approaches for improving generalization involve leveraging 3D perceptions [16, 30, 31, 102, 116] to enhance environmental understanding and using 2D foundation models [11, 60, 79, 111, 119] to enrich the policy's recognition of complex object and scene information. Recent work [47, 121, 122] combines these approaches to further enhance policy performance, though challenges remain in effectively integrating these strategies for broader generalization across diverse manipulation domains.

3 Method

3.1 *AirExo*-2: Efficiently Collecting and Adapting In-the-Wild Demonstrations

As discussed in §2, eliminating the need for physical robots enables low-cost and scalable in-the-wild demonstration collection. However, visual and kinematic discrepancies introduce significant challenges for directly utilizing such data. To overcome this, our objective is to *efficiently convert in-the-wild demonstrations into pseudo-robot demonstrations suitable for training real-world manipulation policies*. This requires **(1)** robust hardware for reliable data collection, **(2)** accurate motion tracking, and **(3)** photo-realistic visual adaptation to support effective policy learning.

To this end, we present the ***AirExo*-2** system, designed to efficiently collect and adapt in-the-wild demonstrations for downstream manipulation policy training, as illustrated in Fig. 2. Unlike prior approaches that *circumvent* the visual gap through in-hand cameras [19, 24, 87, 88, 115] or rely on pre-training and fine-tuning paradigm [17, 23, 29, 50, 100, 104], our system explicitly *bridges* this gap by converting in-the-wild observations into pseudo-robot views using visual adaptors. Combined with accurate motion tracking enabled by the mechanical design, this semi-automatic adaptation process better aligns demonstrations with robot deployment conditions, allowing policies to *directly* transfer to real-world robot platform. Please refer to Appendix A for implementation details.

Hardware. As shown in Fig. 2 (left), we upgrade the AirExo design [29] with aluminum profiles to improve structural robustness and motion tracking precision. To facilitate subsequent demonstration adaptations, we design the ***AirExo*-2** to have a 1:1 scale and maintain kinematic isomorphism with the robotic arm. A mobile base is added to support data collection in diverse environments, and a dedicated calibration process is introduced to ensure accurate motion capture. Details of the

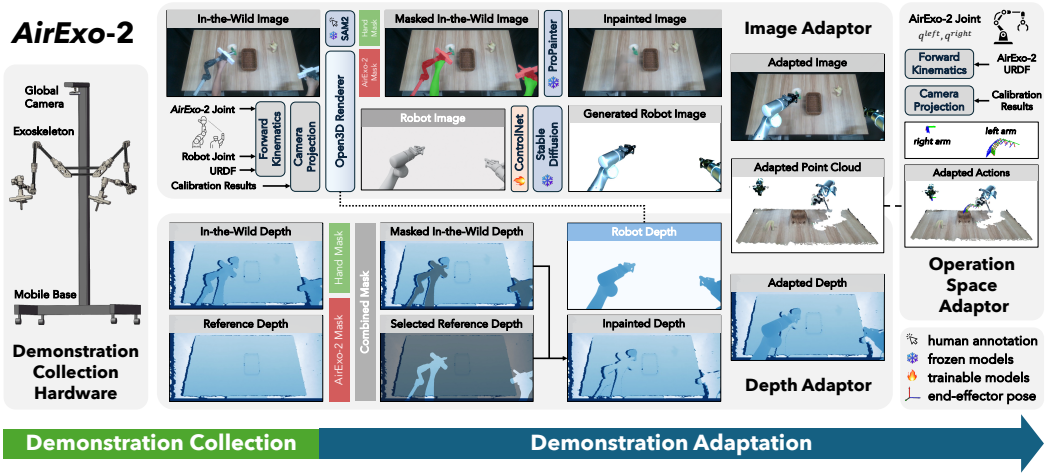


Figure 2: Overview of the *AirExo-2* System. (left) Human uses the low-cost mobile **demonstration collection device** for data collection. (top) The **image adaptor** transforms the RGB images into the robot domain. (bottom) The **depth adaptor** transforms the depth images into the robot domain. (right) The **operation space adaptor** generates the states and actions based on the hardware encoder readings.

hardware design and calibration procedure are provided in Appendix A.1 and Appendix A.2, respectively. The system retains the low-cost advantage of the original AirExo, with the dual-arm platform (excluding the camera) priced at just $600.

Operation Space Adaptor. Following [102, 117], we project all states and actions into the global camera coordinate — a universal coordinate frame between the ***AirExo-2*** and robot platform.

Image Adaptor. As shown in Fig. 2 (top), we convert in-the-wild image into pseudo-robot image by (1) inpainting human-related regions — including hands and ***AirExo-2*** — and (2) overlaying it with the generated robot image. Leveraging the joint encoder readings, calibration results, and the joint mapping between ***AirExo-2*** and the robot, we render RGB-D and mask images of both systems in Open3D [124]. The rendered ***AirExo-2*** mask is merged with the hand mask obtained via SAM-2 [82] to localize human-related areas, which are then inpainted using ProPainter [125]. Inspired by [14], the rendered robot image is refined via ControlNet-guided Stable Diffusion [83, 120] and composited onto the inpainted image to generate the final photo-realistic pseudo-robot image.

Depth Adaptor. As shown in Fig. 2 (bottom), we first capture a universal reference depth of the empty workspace, and combine it with the static object depth from the first frame to form a reference depth for each demonstration. Guided by the combined mask from the image adaptor, we inpaint human-related regions in the depth map with the reference depth to maintain spatial consistency. The adapted depth is obtained by merging the inpainted depth with the rendered robot depth.

3.2 *RISE*-2: A Generalizable Policy for Learning from In-the-Wild Demonstrations

Although in-the-wild demonstrations are adapted into pseudo-robot demonstrations, domain gaps — such as differences in camera viewpoints — still persist. To effectively learn from such data, a generalizable policy is required. As discussed in §2, 3D perception provides view-invariant geometric information, while 2D foundation models offer rich semantic understanding from large-scale pretraining. To leverage the complementary strengths of both, we introduce ***RISE*-2**, a generalizable policy that integrates 3D spatial perception with 2D semantic features for robust robotic manipulation. As illustrated in Fig. 3, the ***RISE*-2** policy consists of a *sparse encoder* for 3D features, a *dense encoder* for 2D features, a *spatial aligner* for feature fusion, and an *action generator* that maps fused features to actions. Implementation details are provided in Appendix B.

Sparse Encoder. 3D point cloud data contains rich spatial structure, enabling effective extraction of local geometric features — a property widely used in general grasping [25, 26, 101]. RISE [102]

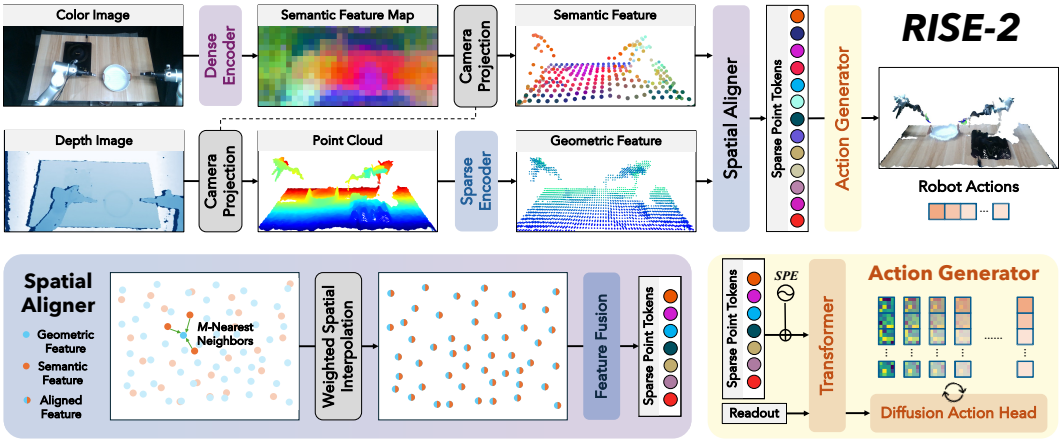


Figure 3: *RISE*-2 Policy Architecture. Given an RGB-D observation as input, the policy predicts continuous actions in the camera frame. The architecture consists of four key components: (1) the **dense encoder** to obtain 2D semantic features; (2) the **sparse encoder** to obtain the 3D local geometric features of seed points; (3) the **spatial aligner** for the feature fusion of 2D semantic features and 3D geometric features guided by their 3D coordinates; (4) the **action generator** to generate continuous robot actions based on the fused feature.

encodes point clouds using raw color information to capture both semantic and geometric cues, but it struggles to disentangle coordinate shifts from color variations. ***RISE*-2** inherits the sparse 3D encoder from RISE, but removes color input to focus purely on geometric features. Formally, the sparse encoder $\mathrm{E}_s : (D, K) \to (\mathbf{F}_g, \mathbf{C}_g)$, maps a depth image D and camera intrinsics K to sparse geometric features $\mathbf{F}_g$, along with the corresponding seed point coordinates $\mathbf{C}_g = \{c_g^i \in \mathbf{P}\}$ after network down-sampling. The encoder adopts a ResNet-like architecture [33] built on MinkowskiEngine [20], enabling efficient sparse feature extraction and supporting real-time policy execution.

Dense Encoder. A generalizable policy requires rich semantic features to understand the scene, while the low-quality texture of point cloud data poses a challenge to this demand. Hence, ***RISE*-2** adopts a dense 2D encoder to focus on semantic understanding based on the color image. Formally, the dense encoder $\mathrm{E}_d : (I, D, K) \to (\mathbf{F}_s, \mathbf{C}_s)$ processes the color image I into semantic feature map $\mathbf{F}_s = \{f_s^i\}$ densely organized in the 2D form of shape $h \times w$, along with its reference 3D coordinates $\mathbf{C}_s = \{c_s^i\}$, calculated by applying 2D adaptive average pooling to size $h \times w$ on the point cloud generated from the depth map D and camera intrinsics K. One key advantage of the dense encoder is its compatibility with 2D foundation models, which provides highly generalizable visual representations across diverse tasks and domains [111]. We implement E_d using DINOv2 [75] fine-tuned with LoRA [42], enhancing the robustness and adaptability of the policy in capturing contextual relationships within the environment.

Spatial Aligner. ***RISE*-2** extracts geometric and semantic features using separate encoders, posing a challenge for feature fusion from different domains. A straightforward solution is to concatenate the aggregated feature vectors, but this discards fine-grained local information critical for precise perception. An alternative [121, 122] upsamples $\mathbf{F}_s$ to match the resolution of the input image I, projects it onto the point cloud, and then downsamples to align with the seed points $\mathbf{C}_g$. However, this process is computationally expensive and reduces the efficiency of policy training and deployment. Instead, ***RISE*-2** utilizes a spatial aligner to efficiently fuse two kinds of features based on their 3D coordinates $\mathbf{C}_g$ and $\mathbf{C}_s$. For each point $c_g^i \in \mathbf{C}_g$ output by the sparse encoder E_s, we compute its M nearest neighbors $\mathbf{N}_i = \{n_1^i, n_2^i, \cdots, n_M^i\} \subseteq \mathbf{C}_s$ from the coordinates output by the dense encoder E_d. The aligned semantic feature f_{s*}^i of c_g^i is computed by weighted spatial interpolation:

$$f_{s*}^i = \frac{\Sigma_{j=1}^{M} f_s^j / \mathrm{dist}(c_g^i, n_j^i)}{\Sigma_{j=1}^{M} 1 / \mathrm{dist}(c_g^i, n_j^i)}, \tag{1}$$

where $\mathrm{dist}(c_i, c_j)$ denotes the Euclidean distance between c_i and c_j. Then, the aligned feature for point c_s^i is obtained by concatenation, *i.e.*, $f_i = \mathrm{Concat}(f_g^i, f_{s*}^i)$. By aligning the seed points $\mathbf{C}_g$

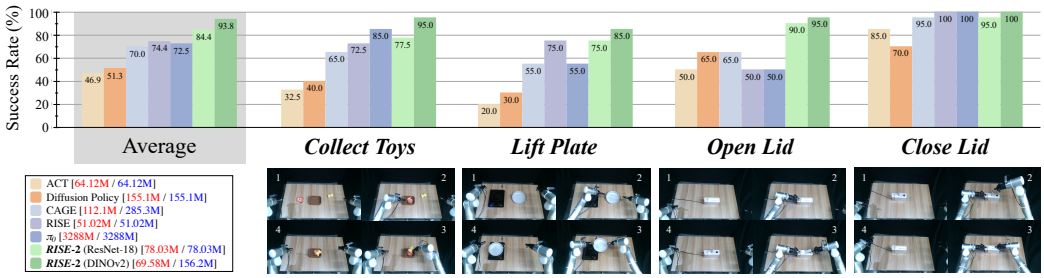


Figure 4: Tasks and In-Domain Evaluation Results. We design 4 tasks to evaluate the performance of the ***RISE*-2** policy. Each policy is noted as "policy [# trainable parameters / # total parameters]". ***RISE*-2** outperforms previous state-of-the-art imitation policies while using relatively few parameters.

with the dense semantic feature map $\mathbf{F}_s$ through their 3D coordinates, we can accurately retrieve fine-grained semantic features at arbitrary locations. The fused features are then aggregated into high-level sparse representations using sparse convolution layers [20]. Visualization of the weighted spatial interpolation applied to the 2D feature map is provided in Appendix B.3.

Action Generator. ***RISE*-2** utilizes a decoder-only transformer [96] to map point tokens with sparse positional encoding [102] to the latent feature, which conditions a diffusion head [18, 37, 46] to generate the action chunk [123]. Actions are represented in the camera frame for cross-scene consistency, with translations in absolute coordinates and rotations in the 6D format [126].

4 Experiments

4.1 Policy Evaluation: *RISE*-2

Setup. To evaluate the performance of policies in both in-domain and generalization settings, we design 4 tasks in Fig. 4 and collect 50 demonstrations for each task using teleoperation [29]. We compare ***RISE*-2** against several representative policies based on 2D images or 3D point clouds, including ACT [123], Diffusion Policy [18], CAGE [111], RISE [102], and π_0 [7]. Additionally, we include a ***RISE*-2** variant with the dense encoder implemented with an ImageNet-pretrained [22] ResNet-18 [33], denoted as ***RISE*-2** (ResNet-18). Please refer to Appendix C.1 for details.

In-Domain Evaluation. As shown in Fig. 4, ***RISE*-2** significantly outperforms previous state-of-the-art imitation policies in average success rate across 4 tasks, demonstrating its effectiveness in handling diverse manipulation challenges. On tasks requiring precise motion execution, such as ***Lift Plate*** and ***Open Lid***, ***RISE*-2** achieves superior accuracy in predicting fine-grained actions, consistently surpassing all baselines. The performance gain over RISE [102] highlights the benefits of using separate 2D and 3D encoders with spatial feature alignment. Further improvements from ***RISE*-2** (ResNet-18) to ***RISE*-2** (DINOv2) underscore the value of leveraging diverse knowledge from visual foundation models [75] in robotic manipulation.

Generalization Evaluation. We select the ***Collect Toys*** task to evaluate the generalization ability of different policies under varying environmental disturbances. Please refer to Appendix C.1 for disturbance details. As shown in Tab. 1, ***RISE*-2** demonstrates strong robustness, maintaining high performance even under combined disturbances. While the ***RISE*-2** (ResNet-18) variant performs slightly below the ***RISE*-2** (DINOv2) model, it still generalizes well and outperforms most baselines. These results further validate our architectural choice of using separate encoders for 3D geometric and 2D semantic features, which effectively enhances policy robustness.

Method	*In-Domain*	*Generalization*		
		Bg.	Obj.	Both
ACT [123]	32.5%	0.0%	2.5%	0.0%
Diffusion Policy [18]	40.0%	12.5%	5.0%	2.5%
CAGE [111]	65.0%	45.0%	42.5%	10.0%
RISE [102]	72.5%	42.5%	12.5%	40.0%
π_0 [7]	85.0%	77.5%	82.5%	**82.5%**
***RISE*-2** (ResNet-18)	77.5%	47.5%	47.5%	37.5%
***RISE*-2** *(DINOv2)*	**95.0%**	**85.0%**	**85.0%**	62.5%

Table 1: Generalization Evaluation Results. "Bg." denotes novel backgrounds and "Obj." denotes novel objects. Policy success rates are reported.

4.2 System Evaluation: *AirExo-2*

Learning from Pseudo-Robot Demonstrations. Previous experiments have demonstrated that ***RISE*-2** is a generalizable policy, making it well-suited for learning from in-the-wild demonstrations collected and processed by ***AirExo*-2**. In this evaluation, we utilize ***AirExo*-2** to collect an equal number of in-the-wild demonstrations and convert them into pseudo-robot demonstrations. These demonstrations are then used to train the ***RISE*-2** policy, which is deployed on a real robot *without further fine-tuning*. As shown in Fig. 5, the policy achieves satisfactory success rates on all tasks, with only a slight performance drop compared to the policy trained on teleoperated demonstrations. This highlights the ability of the ***AirExo*-2** system to collect and adapt high-quality demonstrations suitable for training downstream imitation policies. Please refer to Appendix C.4 for additional results of the RISE policy [102].

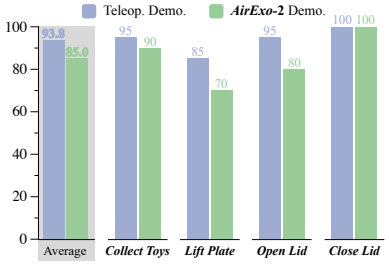
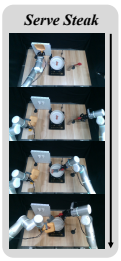


Figure 5: *RISE*-2 Policy Performance with Different Demonstrations. Policies trained solely on demonstrations collected and adapted by ***AirExo*-2** achieve reasonable performance without any robot data, demonstrating the overall effectiveness of the ***AirExo*-2** system.

Complex Task. We evaluate the ***AirExo*-2** system and the ***RISE*-2** policy on the challenging, long-horizon, and contact-rich task ***Serve Steak***. Using 50 in-the-wild demonstrations collected and adapted by ***AirExo*-2**, the ***RISE*-2** policy achieves a decent 50% overall success rate, with strong performance even in the contact-rich *Scoop Steak* phase, as shown in Fig. 6. This demonstrates the effectiveness of ***AirExo*-2** in capturing accurate, fine-grained actions and the ability of ***RISE*-2** to accomplish complex tasks without requiring robot data. Combined with the scalability analysis in §4.3, increasing the number of in-the-wild demonstrations is expected to further boost performance. These results highlight the strong synergy between scalable, high-quality demonstration collection by ***AirExo*-2** and generalizable ***RISE*-2** policy learning, paving the way for efficient and adaptable robotic systems in real-world environments.



Task Phase	Success Rate	
	Teleop.	*AirExo-2*
Pick Plate	100%	90%
Pick Spatula	100%	90%
Scoop Steak	80%	50%
Slide Steak	70%	50%

Figure 6: Results of the *Serve Steak* Task. Trained only with demonstrations by ***AirExo*-2**, the ***RISE*-2** policy achieves decent performance in the challenging task, even without robot data.

Ablation on Visual Adaptors. We use the ***Collect Toys*** task to highlight the importance of the demonstration adaptor for learning from in-the-wild data. As shown in Tab. 2, removing visual adaptors significantly reduces performance, revealing a substantial visual gap between in-the-wild and robot demonstrations. These results also validate the quality of ***AirExo*-2** pseudo-robot demonstrations for training high-performing policies. Notably, ***RISE*-2** without adaptors still performs comparably to RISE [102] with adaptors, demonstrating its strong generalization across embodiments.

Method	Success Rate ↑	
	w.o. adaptors	*w.* adaptors
RISE [102]	30.0%	57.5%
***RISE*-2**	52.5%	90.0%

Table 2: Ablation Results on Visual Adaptors. The visual adaptors help bridge the embodiment visual gap, enabling effective policy transfer from in-the-wild demonstrations.

4.3 System Analysis: *AirExo-2*

User Study. We compare three demonstration collection methods: end-effector pose teleoperation using a haptic device [27], joint-space teleoperation via AirExo [29], and in-the-wild collection with ***AirExo*-2**. Twenty participants (14 men, 6 women, aged 21–35) of varying expertise levels perform the ***Collect Toys*** task on all three platforms *in randomized order*, with each trial consisting of a 3-minute familiarization and a timed demonstration collection. After completing all trials, participants fill out a questionnaire to provide feedback on three collection methods. From the results in Tab. 3, we observe

Method	Completion Time (s) ↓	Avg. Rank ↓		Rating ↑
		Intui.	Learn.	
EE Teleop.	$46.06_{\pm 27.21}$	3.00/3	2.95/3	29.75
Joint Teleop.	$17.31_{\pm 5.055}$	1.80/3	2.00/3	49.58
***AirExo*-2**	$\mathbf{5.66}_{\pm 1.978}$	**1.20**/3	**1.05**/3	**83.00**

Table 3: User Study Results. "Intui." and "Learn." denote intuitiveness and learnability, respectively.

that both experienced and novice users find ***AirExo*-2** easier to learn and operate than both types of teleoperation, leading to faster onboarding and smoother operation. Statistical analysis (Welch's t-test, $p = 8.32 \times 10^{-10}$) further confirms its superior efficiency in task completion time. The user-friendliness and efficiency make ***AirExo*-2** well-suited for diverse users and consistent, high-quality data collection at scale in real-world settings.

Action Accuracy. Accurate actions are essential for learning reliable policies from in-the-wild demonstrations. Unlike handheld devices such as UMI [19], which use visual SLAM for trajectory estimation, ***AirExo*-2** leverages its mechanical structure and forward kinematics for more precise motion tracking. As shown in Tab. 4, it achieves significantly lower translation errors across 3 tracks (please refer to Appendix C.2 for details about accuracy evaluations), confirming its reliability for high-fidelity motion capture.

Device	Error (mm) ↓	
	Avg $\pm$ Std	Max
UMI [19]	$8.855_{\pm 3.228}$	20.002
***AirExo*-2**	$\mathbf{1.737}_{\pm 1.713}$	**6.134**

Table 4: Action Accuracy of Data Collection Systems. ***AirExo*-2** achieves higher action accuracy than handheld devices like UMI [19].

Scalability Analysis. We compare the scalability and effectiveness of ***AirExo*-2** in-the-wild data collection with traditional teleoperation data collection in Fig. 7. Given the same collection time, ***AirExo*-2** yields higher demonstration throughput (analogous to the findings in the previous user study) with lower costs ($0.6k per ***AirExo*-2** system versus approximately $60k per teleoperation platform with robots), making it a more scalable solution to demonstration collection. Moreover, *using the demonstrations collected under the same time*, policies trained on ***AirExo*-2** demonstrations outperform those trained on teleoperation data, underscoring the value of adapted pseudo-robot demonstrations for efficient and effective imitation learning.

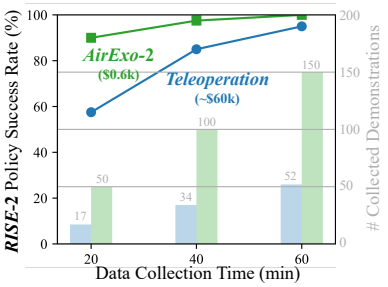


Figure 7: Scalability Analysis Results. ***AirExo*-2** enables scalable and efficient demonstration collection without compromising data effectiveness.

5 Conclusion

This paper introduces ***AirExo*-2**, a novel system designed for scalable in-the-wild demonstration collection and adaptation using low-cost exoskeletons. By incorporating several adaptors, ***AirExo*-2** enables the visual and kinematic transformation of in-the-wild demonstrations into pseudo-robot demonstrations, which can then be directly applied to downstream imitation learning tasks. We also propose a generalizable policy, ***RISE*-2**, which effectively integrates both 2D and 3D perception, demonstrating exceptional performance in both in-domain and generalization scenarios.

Further experiments on a variety of tasks demonstrate that when trained exclusively on pseudo-robot demonstrations generated by the ***AirExo*-2** system — without using any robot data — the policy achieves satisfactory performance during zero-shot deployment on a real-world robot platform, even for the complex, long-horizon, and contact-rich task ***Serve Steak***. This highlights the potential of combining ***AirExo*-2** and ***RISE*-2** as a scalable and promising alternative to traditional teleoperation-imitation pipelines, providing a more efficient, cost-effective solution for scalable, generalizable robotic imitation learning. Together, these results open new possibilities for transferring manipulation skills from in-the-wild environments to real robots, without the need for extensive robot-centric data collection.

6 Limitations and Future Works

While we utilize the proposed demonstration adaptor to visually transform in-the-wild demonstrations collected by ***AirExo*-2** into pseudo-robot demonstrations, these transformed demonstrations are primarily useful for generalizable policies. To enhance the applicability of the pseudo-robot demonstrations to a broader range of policies, future work could explore the integration of demonstration augmentation methods, such as novel view synthesis [14, 85, 94, 95], into the adaptation process to improve the diversity of the demonstrations, making them more versatile for various policy learning.

As demonstrated by several works [19, 41, 54, 115], the in-hand image is a semi-unified observation modality across different embodiments. Our case study in Appendix C.3 also reveals that combining in-hand observations can strengthen the performance of various 2D policies. However, the current ***AirExo*-2** system does not include in-hand cameras. Although we have designed connectors to integrate them with the exoskeleton, calibrating the in-hand cameras with the exoskeleton remains challenging, making it difficult to adapt the in-hand images into the robot domain. Future work could explore effective methods for adapting in-hand images collected by ***AirExo*-2** to the robot domain, or investigate strategies for leveraging the semi-unified in-hand observations in policy design.

The current ***AirExo*-2** system only supports parallel grippers as end-effectors, limiting its applicability in more dexterous tasks. Future work could integrate the ***AirExo*-2** system with dexterous hands [84, 89] and their corresponding exoskeletons [3, 28], enabling more complex manipulation capabilities and expanding the range of tasks the system can effectively perform. Moreover, adding additional sensors to the ***AirExo*-2** system for recording other observation modalities — such as force/torque [34, 63], tactile [61], and sound [64] — can also be explored in the future.

Acknowledgments

We would like to thank Peishen Yan and Chunyu Xue from Shanghai Jiao Tong University for their valuable suggestions on the figures in this paper. We would like to thank Teng Hu from Shanghai Jiao Tong University for insightful discussions about inpainting strategies to mitigate the visual gap.

This work was supported by the National Key Research and Development Project of China (No. 2022ZD0160102), the National Key Research and Development Project of China (No. 2021ZD0110704), Shanghai Artificial Intelligence Laboratory, XPLORER PRIZE grants.

References

[1] Ezra Ameperosa et al. "RoCoDA: Counterfactual Data Augmentation for Data-Efficient Robot Learning from Demonstrations". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2025.

[2] Shikhar Bahl et al. "Affordances from Human Videos as a Versatile Representation for Robotics". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[3] Qingwei Ben et al. "HOMIE: Humanoid Loco-Manipulation with Isomorphic Exoskeleton Cockpit". In: *Robotics: Science and Systems* (2025).

[4] Homanga Bharadhwaj. "Position: Scaling Simulation is Neither Necessary Nor Sufficient for In-the-Wild Robot Manipulation". In: *International Conference on Machine Learning*. 2024.

[5] Homanga Bharadhwaj et al. "Gen2Act: Human Video Generation in Novel Scenarios enables Generalizable Robot Manipulation". In: *arXiv preprint arXiv:2409.16283* (2024).

[6] Homanga Bharadhwaj et al. "RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 4788–4795.

[7] Kevin Black et al. "π_0: A Vision-Language-Action Flow Model for General Robot Control". In: *Robotics: Science and Systems*. 2025.

[8] Gary Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25.11 (2000), pp. 120–123.

[9] Anthony Brohan et al. "RT-1: Robotics Transformer for Real-World Control at Scale". In: *Robotics: Science and Systems*. 2023.

[10] Qingwen Bu et al. "AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems". In: *arXiv preprint arXiv:2503.06669* (2025).

[11] Kaylee Burns et al. "What Makes Pre-Trained Visual Representations Successful for Robust Manipulation?" In: *arXiv preprint arXiv:2312.12444* (2023).

[12] Chi-Lam Cheang et al. "GR-2: A Generative Video-Language-Action Model with Web-Scale Knowledge for Robot Manipulation". In: *arXiv preprint arXiv:2410.06158* (2024).

[13] Jingjing Chen et al. "Towards Effective Utilization of Mixed-Quality Demonstrations in Robotic Manipulation via Segment-Level Selection and Optimization". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2025.

[14] Lawrence Yunliang Chen et al. "RoVi-Aug: Robot and Viewpoint Augmentation for Cross-Embodiment Robot Learning". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 209–233.

[15] Linghao Chen et al. "EasyHeC: Accurate and Automatic Hand-Eye Calibration Via Differentiable Rendering and Space Exploration". In: *IEEE Robotics and Automation Letters* 8.11 (2023), pp. 7234–7241.

[16] Shizhe Chen et al. "PolarNet: 3D Point Clouds for Language-Guided Robotic Manipulation". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1761–1781.

[17] Sirui Chen et al. "ARCap: Collecting High-quality Human Demonstrations for Robot Learning with Augmented Reality Feedback". In: *arXiv preprint arXiv:2410.08464* (2024).

[18] Cheng Chi et al. "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion". In: *Robotics: Science and Systems*. 2023.

[19] Cheng Chi et al. "Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots". In: *Robotics: Science and Systems*. 2024.

[20] Christopher Choy, JunYoung Gwak, and Silvio Savarese. "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3075–3084.

[21] Open X-Embodiment Collaboration et al. "Open X-Embodiment: Robotic Learning Datasets and RT-X Models". In: *IEEE International Conference on Robotics and Automation*. 2024, pp. 6892–6903.

[22] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 248–255.

[23] Jiafei Duan et al. "AR2-D2: Training a Robot Without a Robot". In: *Conference on Robot Learning*. Vol. 229. PMLR, 2023, pp. 2838–2848.

[24] Haritheja Etukuru et al. "Robot Utility Models: General Policies for Zero-Shot Deployment in New Environments". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2025.

[25] Hao-Shu Fang et al. "GraspNet-1billion: A Large-Scale Benchmark for General Object Grasping". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11444–11453.

[26] Hao-Shu Fang et al. "AnyGrasp: Robust and Efficient Grasp Perception in Spatial and Temporal Domains". In: *IEEE Transactions on Robotics* (2023).

[27] Hao-Shu Fang et al. "RH20T: A Comprehensive Robotic Dataset for Learning Diverse Skills in One-Shot". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 653–660.

[28] Hao-Shu Fang et al. "DEXOS: Hand Exoskeleton System for Learning Robot Dexterous Manipulation from Human In-The-Wild". In: *RSS 2025 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*.

[29] Hongjie Fang et al. "AirExo: Low-Cost Exoskeletons for Learning Whole-Arm Manipulation in the Wild". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2024, pp. 15031–15038.

[30] Théophile Gervet et al. "Act3D: 3D Feature Field Transformers for Multi-Task Robotic Manipulation". In: *Conference on Robot Learning*. Ed. by Jie Tan, Marc Toussaint, and Kourosh Darvish. Vol. 229. PMLR, 2023, pp. 3949–3965.

[31] Ankit Goyal et al. "RVT: Robotic View Transformer for 3D Object Manipulation". In: *Conference on Robot Learning*. PMLR. 2023, pp. 694–710.

[32] Haoran He et al. "Large-Scale Actionless Video Pre-Training via Discrete Diffusion for Efficient Policy Learning". In: *arXiv preprint arXiv:2402.14407* (2024).

[33] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[34] Zihao He et al. "FoAR: Force-Aware Reactive Policy for Contact-Rich Robotic Manipulation". In: *IEEE Robotics and Automation Letters* (2025).

[35] Tom Henighan et al. "Scaling Laws for Autoregressive Generative Modeling". In: *arXiv preprint arXiv:2010.14701* (2020).

[36] Nick Heppert et al. "DITTO: Demonstration Imitation by Trajectory Transformation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2024, pp. 7565–7572.

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[38] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *arXiv preprint arXiv:2207.12598* (2022).

[39] Zhengdong Hong, Kangfu Zheng, and Linghao Chen. "Fully Automatic Hand-Eye Calibration with Pretrained Image Models". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2024).

[40] Ryan Hoque et al. "IntervenGen: Interventional Data Generation for Robust and Data-Efficient Robot Imitation Learning". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2024, pp. 2840–2846.

[41] Kyle Hsu et al. "Vision-Based Manipulators Need to Also See from Their Hands". In: *International Conference on Learning Representations*. 2022.

[42] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2022.

[43] Pu Hua et al. "GenSim2: Scaling Robot Data Generation with Multi-Modal and Reasoning LLMs". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 5030–5066.

[44] Andrew Jaegle et al. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". In: *International Conference on Learning Representations*. 2022.

[45] Eric Jang et al. "BC-Z: Zero-Shot Task Generalization with Robotic Imitation Learning". In: *Conference on Robot Learning*. Vol. 164. PMLR, 2021, pp. 991–1002.

[46] Michael Janner et al. "Planning with Diffusion for Flexible Behavior Synthesis". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 9902–9915.

[47] Yueru Jia et al. "Lift3D Foundation Policy: Lifting 2D Large-Scale Pretrained Models for Robust 3D Robotic Manipulation". In: *arXiv preprint arXiv:2411.18623* (2024).

[48] Zhenyu Jiang et al. "DexMimicGen: Automated Data Generation for Bimanual Dexterous Manipulation via Imitation Learning". In: *arXiv preprint arXiv:2410.24185* (2024).

[49] Jared Kaplan et al. "Scaling Laws for Neural Language Models". In: *arXiv preprint arXiv:2001.08361* (2020).

[50] Simar Kareer et al. "EgoMimic: Scaling Imitation Learning via Egocentric Video". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2025.

[51] Hiroharu Kato et al. "Differentiable Rendering: A Survey". In: *arXiv preprint arXiv:2006.12057* (2020).

[52] Alexander Khazatsky et al. "DROID: A Large-Scale In-the-Wild Robot Manipulation Dataset". In: *Robotics: Science and Systems*. 2024.

[53] Heecheol Kim et al. "Training Robots Without Robots: Deep Imitation Learning for Master-to-Robot Policy Transfer". In: *IEEE Robotics and Automation Letters* 8.5 (2023), pp. 2906–2913.

[54] Moo Jin Kim, Jiajun Wu, and Chelsea Finn. "Giving Robots a Hand: Learning Generalizable Manipulation with Eye-in-Hand Human Video Demonstrations". In: *arXiv preprint arXiv:2307.05959* (2023).

[55] Moo Jin Kim et al. "OpenVLA: An Open-Source Vision-Language-Action Model". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 2679–2713.

[56] Diederik P Kingma. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[57] Marion Lepert, Jiaying Fang, and Jeannette Bohg. "Phantom: Training Robots Without Robots Using Only Human Videos". In: *arXiv preprint arXiv:2503.00779* (2025).

[58] Tzu-Mao Li et al. "Differentiable Monte Carlo Ray Tracing through Edge Sampling". In: *ACM Transactions on Graphics* 37.6 (2018), 222:1–222:11.

[59] Fanqi Lin et al. "Data Scaling Laws in Imitation Learning for Robotic Manipulation". In: *International Conference on Learning Representations*. 2025.

[60] Xingyu Lin et al. "SpawnNet: Learning Generalizable Visuomotor Skills from Pre-trained Network". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 4781–4787.

[61] Fangchen Liu et al. "ViTaMIn: Learning Contact-Rich Tasks Through Robot-Free Visuo-Tactile Manipulation Interface". In: *arXiv preprint arXiv:2504.06156* (2025).

[62] Songming Liu et al. "RDT-1B: A Diffusion Foundation Model for Bimanual Manipulation". In: *International Conference on Learning Representations*. 2025.

[63] Wenhai Liu et al. "ForceMimic: Force-Centric Imitation Learning with Force-Motion Capture System for Contact-Rich Manipulation". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2025.

[64] Zeyi Liu et al. "ManiWAV: Learning Robot Manipulation from In-the-Wild Audio-Visual Data". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 947–962.

[65] Shengcheng Luo et al. "Human-Agent Joint Learning for Efficient Robot Manipulation Skill Acquisition". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2025.

[66] Jun Lv et al. "SAM-RL: Sensing-Aware Model-Based Reinforcement Learning via Differentiable Physics-Based Simulation and Rendering". In: *Robotics: Science and Systems*. 2023.

[67] Yecheng Jason Ma et al. "LIV: Language-Image Representations and Rewards for Robotic Control". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 23301–23320.

[68] Yecheng Jason Ma et al. "VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training". In: *International Conference on Learning Representations*. 2023.

[69] Arjun Majumdar et al. "Where are We in the Search for an Artificial Visual Cortex for Embodied Intelligence?" In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 655–677.

[70] Ajay Mandlekar et al. "MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations". In: *Conference on Robot Learning*. Vol. 229. PMLR, 2023, pp. 1820–1864.

[71] Robert McCarthy et al. "Towards Generalist Robot Learning from Internet Video: A Survey". In: *arXiv preprint arXiv:2404.19664* (2024).

[72] Yao Mu et al. "RoboTwin: Dual-Arm Robot Benchmark with Generative Digital Twins". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025, pp. 27649–27660.

[73] Suraj Nair et al. "R3M: A Universal Visual Representation for Robot Manipulation". In: *Conference on Robot Learning*. PMLR. 2022, pp. 892–909.

[74] Octo Model Team et al. "Octo: An Open-Source Generalist Robot Policy". In: *Robotics: Science and Systems*. 2024.

[75] Maxime Oquab et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *Transactions on Machine Learning Research* 2024 (2024).

[76] Georgios Papagiannis et al. "R+X: Retrieval and Execution from Everyday Human Videos". In: *arXiv preprint arXiv:2407.12957* (2024).

[77] William Peebles and Saining Xie. "Scalable Diffusion Models with Transformers". In: *IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4195–4205.

[78] Dean Pomerleau. "ALVINN: An Autonomous Land Vehicle in a Neural Network". In: *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 1988, pp. 305–313.

[79] Jianing Qian, Anastasios Panagopoulos, and Dinesh Jayaraman. "Recasting Generic Pretrained Vision Transformers As Object-Centric Scene Encoders For Manipulation Policies". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 17544–17552.

[80] Yuzhe Qin et al. "DexMV: Imitation Learning for Dexterous Manipulation from Human Videos". In: *European Conference on Computer Vision*. Springer. 2022, pp. 570–587.

[81] Ilija Radosavovic et al. "Real-World Robot Learning with Masked Visual Pre-Training". In: *Conference on Robot Learning*. PMLR. 2022, pp. 416–426.

[82] Nikhila Ravi et al. "SAM 2: Segment Anything in Images and Videos". In: *International Conference on Learning Representations*. 2025.

[83] Robin Rombach et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[84] Branden Romero et al. "EyeSight Hand: Design of a Fully-Actuated Dexterous Robot Hand with Integrated Vision-Based Tactile Sensors and Compliant Actuation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2024, pp. 1853–1860.

[85] Kyle Sargent et al. "ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9420–9429.

[86] Johannes L Schonberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4104–4113.

[87] Mingyo Seo et al. "LEGATO: Cross-Embodiment Imitation Using a Grasping Tool". In: *IEEE Robotics and Automation Letters* 10.3 (2025), pp. 2854–2861.

[88] Nur Muhammad Mahi Shafiullah et al. "On Bringing Robots Home". In: *arXiv preprint arXiv:2311.16098* (2023).

[89] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. "LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning". In: *Robotics: Science and Systems*. 2023.

[90] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. "Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation". In: *Conference on Robot Learning*. Ed. by Karen Liu, Dana Kulic, and Jeffrey Ichnowski. Vol. 205. PMLR, 2022, pp. 785–799.

[91] Laura M. Smith et al. "AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos". In: *Robotics: Science and Systems*. 2020.

[92] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2021.

[93] Mohan Kumar Srirama et al. "HRP: Human Affordances for Robotic Pre-Training". In: *Robotics: Science and Systems*. 2024.

[94] Stephen Tian et al. "View-Invariant Policy Learning via Zero-Shot Novel View Synthesis". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 1173–1193.

[95] Basile Van Hoorick et al. "Generative Camera Dolly: Extreme Monocular Dynamic Novel View Synthesis". In: *European Conference on Computer Vision*. Springer. 2025, pp. 313–331.

[96] Ashish Vaswani et al. "Attention is All You Need". In: *Advances in Neural Information Processing Systems* 30 (2017).

[97] Vitalis Vosylius and Edward Johns. "Instant Policy: In-Context Imitation Learning via Graph Diffusion". In: *International Conference on Learning Representations*. 2025.

[98] Homer Rich Walke et al. "BridgeData V2: A Dataset for Robot Learning at Scale". In: *Conference on Robot Learning*. Vol. 229. PMLR, 2023, pp. 1723–1736.

[99] Chen Wang et al. "MimicPlay: Long-Horizon Imitation Learning by Watching Human Play". In: *Conference on Robot Learning*. 2023, pp. 201–221.

[100] Chen Wang et al. "DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation". In: *Robotics: Science and Systems*. 2024.

[101] Chenxi Wang et al. "Graspness Discovery in Clutters for Fast and Accurate Grasp Detection". In: *IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15964–15973.

[102] Chenxi Wang et al. "RISE: 3D Perception Makes Real-World Robot Imitation Simple and Effective". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2024, pp. 2870–2877.

[103] Jun Wang et al. "CyberDemo: Augmenting Simulated Human Demonstration for Real-World Dexterous Manipulation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2024, pp. 17952–17963.

[104] Jun Wang et al. "EVE: Enabling Anyone to Train Robots using Augmented Reality". In: *ACM Symposium on User Interface Software and Technology*. 2024, pp. 1–13.

[105] Lirui Wang et al. "GenSim: Generating Robotic Simulation Tasks via Large Language Models". In: *International Conference on Learning Representations*. 2024.

[106] Lirui Wang et al. "Scaling Proprioceptive-Visual Learning with Heterogeneous Pre-Trained Transformers". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 124420–124450.

[107] Yufei Wang et al. "RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation". In: *International Conference on Machine Learning*. 2024.

[108] Chuan Wen et al. "Any-point Trajectory Modeling for Policy Learning". In: *Robotics: Science and Systems*. 2024.

[109] Hongtao Wu et al. "Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation". In: *International Conference on Learning Representations*. 2024.

[110] Kun Wu et al. "RoboMIND: Benchmark on Multi-embodiment Intelligence Normative Data for Robot Manipulation". In: *Robotics: Science and Systems*. 2025.

[111] Shangning Xia et al. "CAGE: Causal Attention Enables Data-Efficient Generalizable Robotic Manipulation". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2025.

[112] Haoyu Xiong et al. "Learning by Watching: Physical Imitation of Manipulation Skills from Human Videos". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2021, pp. 7827–7834.

[113] Mengda Xu et al. "Flow as the Cross-Domain Manipulation Interface". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 2475–2499.

[114] Seonghyeon Ye et al. "Latent Action Pretraining from Videos". In: *International Conference on Learning Representations*. 2025.

[115] Sarah Young et al. "Visual Imitation Made Easy". In: *Conference on Robot Learning*. Vol. 155. PMLR, 2020, pp. 1992–2005.

[116] Yanjie Ze et al. "3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations". In: *Robotics: Science and Systems*. 2024.

[117] Yanjie Ze et al. "Generalizable Humanoid Manipulation with Improved 3D Diffusion Policies". In: *arXiv preprint arXiv:2410.10803* (2024).

[118] Jia Zeng et al. "Learning Manipulation by Predicting Interaction". In: *Robotics: Science and Systems*. 2024.

[119] Junjie Zhang et al. "SAM-E: Leveraging Visual Foundation Model with Sequence Imitation for Embodied Manipulation". In: *International Conference on Machine Learning*. 2024.

[120] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. "Adding Conditional Control to Text-to-Image Diffusion Models". In: *IEEE/CVF International Conference on Computer Vision*. 2023, pp. 3836–3847.

[121] Tong Zhang et al. "A Universal Semantic-Geometric Representation for Robotic Manipulation". In: *Conference on Robot Learning*. Vol. 229. PMLR, 2023, pp. 3342–3363.

[122] Tong Zhang et al. "Leveraging Locality to Boost Sample Efficiency in Robotic Manipulation". In: *Conference on Robot Learning*. Vol. 270. PMLR, 2024, pp. 3264–3284.

[123] Tony Z Zhao et al. "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware". In: *Robotics: Science and Systems*. 2023.

[124] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: *arXiv preprint arXiv:1801.09847* (2018).

[125] Shangchen Zhou et al. "ProPainter: Improving Propagation and Transformer for Video Inpainting". In: *IEEE/CVF International Conference on Computer Vision*. 2023, pp. 10477–10486.

[126] Yi Zhou et al. "On the Continuity of Rotation Representations in Neural Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5745–5753.

[127] Brianna Zitkovich et al. "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control". In: *Conference on Robot Learning*. Vol. 229. PMLR, 2023, pp. 2165–2183.

Appendix

A *AirExo*-2 Implementation

A.1 Hardware Design

High-quality in-the-wild data requires precise collection for effective pseudo-robot transformation. However, AirExo [29] has several key hardware limitations:

- **H1**. Most of its components are 3D-printed using polylactic acid (PLA), leading to low rigidity and susceptibility to structural deformation.
- **H2**. Unlike typical robots with constrained ranges, its joints can rotate beyond 360°, potentially reaching positions the robot cannot achieve.
- **H3**. Although portable, inevitable body movements during operations can cause its base to shift, leading to inaccurate action recording.
- **H4**. Its gripper lacks smooth control, leading to potential jamming under clamping forces.
- **H5**. The shaft connects directly to the encoder, with side-routed wires that endure friction and stretching during joint movement, risking long-term damage or short circuits.

During demonstration collection using teleoperation, most of the above issues (except **H4**) might not significantly impact data acquisition, as the human operator can adjust their actions based on the movements of the robotic arms. However, during the in-the-wild demonstration collection, these drawbacks can substantially affect motion capture accuracy and may result in invalid or unusable demonstration data.

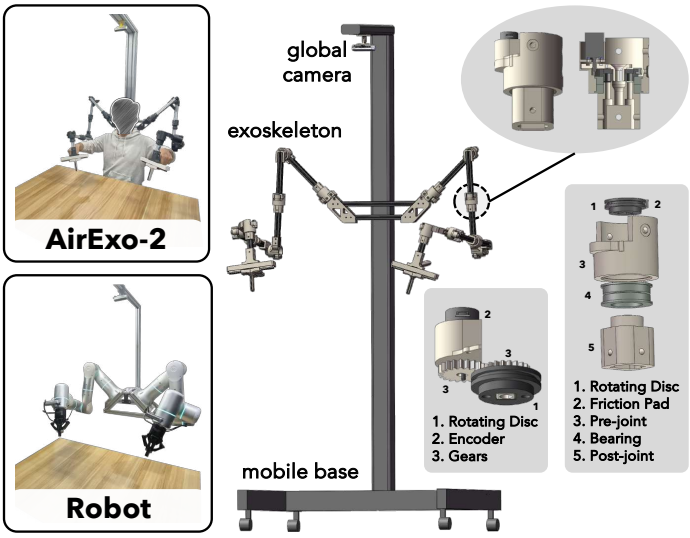


Figure 8: Hardware Design of *AirExo*-2. The ***AirExo*-2** demonstration collection platform consists of a mobile base and a dual-arm exoskeleton. Global cameras can be mounted on top of the platform to capture visual observations during data collection. The detailed joint structure is shown on the right side, featuring two key designs: hollow rotating discs and side-mounted encoders; joints with angle limit and adjustable friction.

Driven by the limitations above, the hardware design of ***AirExo*-2** focuses on seamless integration with the learning process from in-the-wild demonstrations. To enable direct learning, the exoskeleton is built with a 1:1 dimensional match to the robotic arm, minimizing discrepancies between human and robot embodiments. An overview of the hardware is shown in Fig. 2. Excluding cameras, the complete ***AirExo*-2** platform costs $600. Key hardware design elements are outlined below.

Enhanced Overall Structural Rigidity (H1). In AirExo, the links connecting two consecutive joints are the most prone to deformation. In ***AirExo*-2**, we replace the 3D-printed parts with 20x20

European standard aluminum profiles, providing significantly higher strength at a very low cost. For the joints, the outer shell is 3D-printed using PLA-CF, a carbon fiber-reinforced PLA material with higher hardness. Inside the joint, larger bearings are used to further enhance structural rigidity. Together with the improved links, the hardware upgrade significantly increases the overall structural rigidity of ***AirExo*-2**, making the exoskeleton more durable, and thereby improving system accuracy.

Hollow Rotating Disc and Side-Mounted Encoder (H5). As shown in Fig. 8, this design features an encoder mounted on the side of the joint, which uses a gear mechanism to translate the rotational angle of the rotating disc into encoder readings. The hollow disc design allows wires to pass through, preventing them from stretching during rotation and thereby extending their lifespan. Additionally, the side-mounted encoder simplifies maintenance, enabling easy debugging and replacement without the need to disassemble the joint.

Joint with Angle Limit and Adjustable Friction (H2). This structure consists of a rotating disc with a grooved track and a friction pad that can be embedded into the track. Together, they allow for adjustment of the limiting angle, ensuring that the motion range of the ***AirExo*-2** joint exactly aligns with the corresponding joint range of the robot. As illustrated in Fig. 8, the rotating disc, pre-joint, and post-joint are connected through bearings, allowing the rotational motion of the joint to be directly transmitted to the track. For demonstration collection, excessive or insufficient friction in the rotation of the joints is undesirable. Hence, the friction force of the joint in ***AirExo*-2** can be adjusted by turning the screw on the outer shell, which compresses the friction pad. This design ensures optimal friction for comfortable and accurate data collection.

Smooth Gripper Control (H4). Following [19, 63], the gripper of ***AirExo*-2** incorporates a linear guide, with the fingers mounted on a sliding block. This design allows for smoother opening and closing of the gripper, ensuring it operates seamlessly under clamping forces without any stalling.

Mobile Data Collection Platform (H3). Portability is crucial for in-the-wild data collection. However, to address the issue of base movement caused by the body motion of the operator, we mount ***AirExo*-2** on a mobile aluminum profile stand, as shown in Fig. 8. This setup ensures stability of the base during demonstration collection while maintaining the flexibility needed for mobility, enabling large-scale demonstration collection in real-world environments. An Intel RealSense D415 camera is set up on the top of the mobile platform to capture global observations. We also designed two optional camera mounts (though not used in this paper) for the future integration of in-hand cameras on the top of both grippers.

A.2 Calibration

The ***AirExo*-2** system requires two types of calibration simultaneously: (1) aligning the zero positions of each joint with the corresponding robot joint, and (2) determining the transformation between the global camera and the ***AirExo*-2** base. To address these challenges, we propose a two-stage calibration process.

Initial Calibration. For initial calibration, the former calibration can be achieved by manually adjusting the joints to approximate the zero position using specialized 3D-printed tools and reading the encoder values, obtaining $\{\tilde{q}^{\text{left}}_{\text{calib}}, \tilde{q}^{\text{right}}_{\text{calib}}\}$. The latter calibration can be done by attaching an ArUco calibration marker board with a known position on the base $\mathbf{T}^{\text{base}}_{\text{marker}}$ and performing optical calibration using OpenCV [8], obtaining $\mathbf{T}^{\text{camera}}_{\text{marker}}$. Thus, the transformation can calculated as

$$\left[\tilde{\mathbf{t}}^{\text{camera}}_{\text{base}} \mid \tilde{\mathbf{r}}^{\text{camera}}_{\text{base}}\right] \stackrel{\text{def}}{=} \tilde{\mathbf{T}}^{\text{camera}}_{\text{base}} = \mathbf{T}^{\text{camera}}_{\text{marker}} \left(\tilde{\mathbf{T}}^{\text{base}}_{\text{marker}}\right)^{-1} \tag{2}$$

However, this approach introduces errors due to human observation, calibration board misalignment, and optical inaccuracies. In a chained system like ***AirExo*-2**, these errors may propagate and amplify across joints, leading to significant cumulative inaccuracies in the end-effector pose. Therefore, fine-grained calibration is essential to ensure precise and consistent alignment between ***AirExo*-2** and the camera frames during demonstration collection.

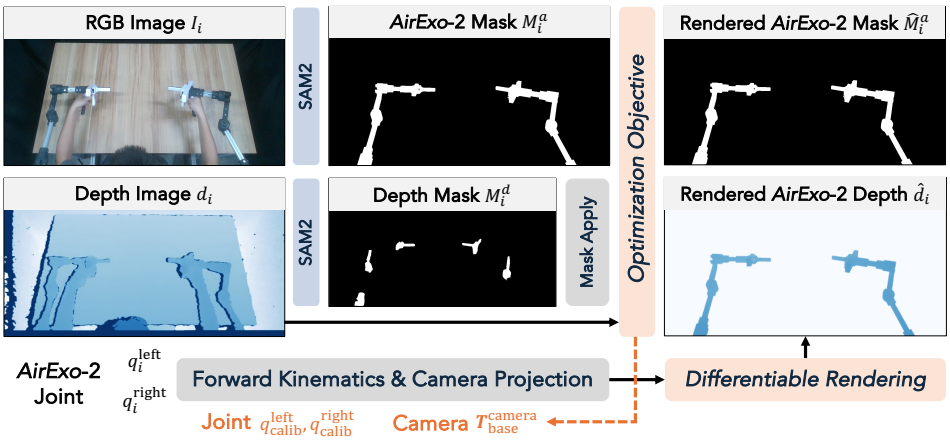


Figure 9: Calibration via Differentiable Rendering. The parameters in orange denote the calibration parameters to be optimized via differentiable rendering.

Calibration via Differentiable Rendering. In the second stage, inspired by prior works [15, 39, 66], we use differentiable rendering [51] to refine the initial calibration. Training samples are obtained from a single human *play* trajectory with ***AirExo-2***. Using the joint states and calibration parameters, we render the system mask and depth via a differentiable rendering engine [58]. Calibration parameters are optimized by minimizing discrepancies between the rendered and annotated system masks, as well as between the rendered and observed depths. Pseudo-ground-truth masks are manually annotated with SAM-2 [82]. This iterative refinement compensates for errors that accumulate across joints, ultimately improving the overall accuracy.

Specifically, we define p, the calibration parameters to be optimized, as

$$p \stackrel{\text{def}}{=} \{\Delta\mathbf{t}_{\text{base}}^{\text{camera}}, \mathbf{r}_{\text{base}}^{\text{camera}}, \Delta q_{\text{calib}}^{\text{left}}, \Delta q_{\text{calib}}^{\text{right}}\} \tag{3}$$

where parameters, except for the base-to-camera rotation, are represented as deltas relative to the initial calibration results. The base-to-camera rotation is expressed in a 6D format [126]. Thus, the final calibration results can be calculated as

$$\mathbf{T}_{\text{base}}^{\text{camera}} = \left[\tilde{\mathbf{t}}_{\text{base}}^{\text{camera}} + \Delta\mathbf{t}_{\text{base}}^{\text{camera}} \mid \mathbf{r}_{\text{base}}^{\text{camera}}\right], \tag{4}$$

$$q_{\text{calib}}^{\text{type}} = \tilde{q}_{\text{calib}}^{\text{type}} + \Delta q_{\text{calib}}^{\text{type}}, \quad \text{type} \in [\text{left}, \text{right}], \tag{5}$$

and the initial parameter values are set as

$$p_0 = \{\mathbf{0}, \tilde{r}_{\text{base}}^{\text{camera}}, \mathbf{0}, \mathbf{0}\} \tag{6}$$

For the optimization process, we first record a single in-the-wild *play* trajectory, in which the human operator uses the ***AirExo-2*** to adopt various poses. During trajectory recording, ensure that all parts of the ***AirExo-2*** remain above the human hands and arms from the camera's perspective. After data collection, we sample approximately 40 image-joint pairs from the trajectory, denoted as $\{I_i, d_i, q_i^{\text{left}}, q_i^{\text{right}}\}_{i=1}^{N_c}$, where N_c represents the total number of training samples for calibration, and I_i and d_i are the RGB and depth images of the i-th sample, respectively. Subsequently, we utilize SAM-2 [82] to annotate the ***AirExo-2*** mask M_i^a and the depth mask $M_i^d \subseteq M_i^a$ for each sample i, as shown in Fig. 9. The first mask provides supervision for the rendered ***AirExo-2*** mask, while the second mask is used to select the valid depth information that serves as the supervision signal.

The differentiable rendering engine Redner [58] is employed to render the ***AirExo-2*** mask $\hat{M}_i^a$ and ***AirExo-2*** depth $\hat{d}_i$ using the calibration results and joint information:

$$\hat{M}_i^a, \hat{d}_i = \mathscr{R}(p; q_i^{\text{left}}, q_i^{\text{right}}, \tilde{T}_{\text{base}}^{\text{camera}}, \tilde{q}^{\text{left}}, \tilde{q}^{\text{right}}), \tag{7}$$

where the rendering engine $\mathscr{R}(p; \cdots)$ computes the gradients of the calibration parameters p during the rendering process, and $\tilde{T}_{\text{base}}^{\text{camera}}, \tilde{q}^{\text{left}}, \tilde{q}^{\text{right}}$ represent the initial calibration results.

The rendered ***AirExo-2*** mask $\hat{M}_i^a$ is supervised by the human-annotated pseudo-***AirExo-2*** mask M_i^a, and the rendered ***AirExo-2*** depth $\hat{d}_i$ is supervised by the camera depth d_i within the region of the

human-annotated depth mask M_i^d. The depth mask ensures that only accurate depth information contributes to the loss. Thus, the objective can be written as:

$$\mathscr{L}(p) = \frac{1}{N_c} \sum_{i=1}^{N_c} \left(\beta \cdot \left\| M_i^a - \hat{M}_i^a \right\|^2 + \left\| d_i - \hat{d}_i \right\|^2 \circ M_i^d \right), \tag{8}$$

where β represents the weighting coefficient, and $\circ$ denotes the mask-apply operation. In practice, we set $\beta = 5$, use $N_c = 40$ samples for optimization, and employ the Adam optimizer [56] with a learning rate of 10^{-4} for 1000 iterations to fine-tune the calibration results.

Calibration Analysis. Precise calibration is vital for accurate action tracking. In addition to our proposed full two-stage calibration method, we evaluate three alternatives: (1) *initial calibration*, using first-stage results without refinement; (2) *human annotation*, where a user manually adjusts calibration parameters via a real-time GUI; and (3) *two-stage calibration (mask only)*, which refines calibration using only mask differences as supervision. As shown in Tab. 5, our proposed full two-stage method achieves the highest accuracy, with a 0.78% mask difference and 14.0 mm depth error. While depth errors are affected by sensor noise and cannot fully reflect action accuracy (please refer to §4.3 for action accuracy analysis), the results confirm the benefits of modeling 3D information through differentiable rendering for accurate calibration. Notably, human annotation even performs worse than initial calibration due to its reliance on 2D visual cues, which limits depth estimation accuracy.

Calibration Method	**Difference** ↓	
	Mask (%)	**Depth** (mm)
Initial Calibration	$1.71_{\pm 0.37}$	$21.6_{\pm 5.2}$
Human Annotation	$2.31_{\pm 0.31}$	$31.2_{\pm 6.4}$
Two-Stage Calibration (mask only)	$1.10_{\pm 0.26}$	$17.6_{\pm 4.1}$
Two-Stage Calibration (mask + depth)	$\mathbf{0.78}_{\pm 0.25}$	$\mathbf{14.0}_{\pm 2.9}$

Table 5: Calibration Analysis Results. Using our proposed two-stage calibration, we achieve higher accuracy than both initial calibration and human annotations. Including depth as additional supervision also helps the optimization process convergence.

A.3 Visual Adaptors

Semi-Automatic SAM-2 Annotations. In the image adaptor, we initially annotate the hand mask (and, if visible, the head mask) manually using SAM-2 [82]. After a few annotations on approximately 50 scenes, we can fine-tune SAM-2 on human-annotated samples, enabling automated labeling. This significantly reduces human effort and streamlines the demonstration adaptation process, making it nearly fully automated.

ControlNet Training. We train a ControlNet [120] based on the Stable Diffusion 1.5 [83] model to generate photo-realistic robot images from rendered robot images. To collect training samples, we use teleoperation to gather a small amount of *play* data, where the robot is teleoperated to move randomly within an empty workspace while recording RGB-D images and corresponding joint states. This ensures a diverse dataset of robot arm configurations, free from occlusions or distractions.

Notably, these training samples are ***platform-specific*** but ***task-invariant***, meaning they only need to be collected once per robot platform, and the trained ControlNet can be used across all tasks. This also opens up the possibility of directly transforming our in-the-wild demonstrations to other robotic arms without the need to design new exoskeletons that match specific robots.

For training, we use a batch size of 88 and a learning rate of 10^{-5}, while keeping other hyperparameters at their default settings. We use 50 DDPM sampling steps [37] with a guidance scale of 9.0 [38]. The prompt for generating robot images for our robot platform is displayed as follows.

```
Prompt: robotic arms, dual arm, industrial robotic manipulator,
metallic silver color, mechanical joints, precise mechanical details,
gripper end effector, high-quality photo, photorealistic, clear and
sharp details
```

A.4 Discussions

Hardware Improvements over AirExo [29]. We improved both the *repairability* and *accuracy* of the system to enable scalable, high-quality data collection. In terms of repairability, relocating

the encoders outside the joint and adopting a gear-driven design reduced encoder replacement time from 15 minutes to just 2 minutes. In terms of accuracy, the system's action recording precision was significantly enhanced, as presented in Tab. 4.

Computation Overhead of Adaptation. Rendering a 1-minute scene of approximately 600 frames requires about 40 seconds on a single GPU, with the majority of the time consumed by the ControlNet-tuned Stable Diffusion [83, 120] model. This process can be further accelerated by: (1) rendering at a lower resolution, and (2) leveraging multiple GPUs in parallel. Notably, *the primary bottleneck in data collection lies in human effort rather than computation*. Given that computational resources can be scaled more readily, we argue that prioritizing the reduction of human involvement — even at the cost of slightly increased computational usage — can significantly improve the overall efficiency and scalability of the data collection pipeline.

Cross-Robot Transfer and Data Reusability. Our adaptation pipeline can be easily extended to other robots by deriving joint values from end-effector poses via inverse kinematics, followed by rendering and inpainting the corresponding robot. As shown in Tab. 2, even without visual adaptation, policies trained solely on in-the-wild demonstrations can sometimes complete tasks successfully. This suggests that raw in-the-wild data itself can already be used to train policies and deployed on different robots. By incorporating robot-specific adaptations, we can generate high-quality data for different robots and boost policy performance by training on these adapted demonstrations. This approach can greatly enhance data reusability across platforms and embodiments.

Strict 1:1 Joint Mapping Design of *AirExo*-2. In our experiments, the Flexiv arms are comparable in size to the commonly used robotic arms like Franka, KUKA, and xArms. As shown in Tab. 3, the user study indicates that operators find our system intuitive and easy to use, suggesting that exact 1:1 scaling is not a prerequisite for effective data collection. Moreover, as most humanoid robots are designed to match human dimensions, our system is naturally compatible. For manipulators of different sizes, human-sized exoskeletons can still be employed, with data transfer facilitated through the adaptation process described in the previous discussion.

B *RISE*-2 Implementation

B.1 Data Processing

Image. The color image is resized to 448×252 for DINOv2 backbone [75] and 640×360 for ResNet-18 backbone [33]. The depth image is resized to 640×360 before creating the point cloud. The camera intrinsics are adjusted accordingly. Both the point clouds and actions are projected to the camera coordinate system following [102, 117].

Point Cloud. The point cloud is down-sampled with a voxel size of 5mm. For the data collected with teleoperation, we crop the point clouds using the range of

$$(x,y,z) \in [-0.70\text{m}, 0.70\text{m}] \times [-0.30\text{m}, 0.55\text{m}] \times [0.90\text{m}, 1.55\text{m}].$$

For the data collected with ***AirExo*-2**, we crop the point clouds using the range of

$$(x,y,z) \in [-0.70\text{m}, 0.70\text{m}] \times [-0.30\text{m}, 0.45\text{m}] \times [0.75\text{m}, 1.4\text{m}].$$

Robot Trajectory Sampling. The robot trajectories are sampled using differences of translation, rotation, and gripper width to remove redundant actions. For the action at two adjacent timesteps, if all the differences are less than the thresholds, only the first action is retained. The threshold for translation and gripper width is 5mm, and the rotation threshold is $\pi/24$.

B.2 Hyperparameters

Policy. The sparse encoder adopts a ResNet-like architecture built upon MinkowskiEngine [20]. The dense encoder adopts DINOv2-base [75] as the 2D backbone with the output channel of 128. In the spatial aligner, we use $M = 3$ for feature alignment. The aligned features are fused by shared MLPs with the size of (256, 256, 256), and then fed into another sparse network. The two sparse

networks are detailed in Tab. 6. The transformer in action generator contains 4 blocks, in which we set $d_{\text{model}} = 512$ and $d_{\text{ff}} = 2048$ following [102]. The channel number of the readout token is 512. The diffusion head adopts a CNN implementation [18] with 100 denoising iterations in training and 20 iterations in inference. The output action horizon used in experiments is 20.

Layer Name	Sparse Encoder	Spatial Aligner
init_conv	$k=[3,3,3]$, $c=32$, $d=1$, $s=1$ 2x mean pooling	-
conv1	$k=[3,3,3]$, $c=32$, $d=1$, $s=1$	$k=[3,3,3]$, $c=256$, $d=4$, $s=4$
conv2	$k=[3,3,3]$, $c=64$, $d=2$, $s=1$	$k=[3,3,3]$, $c=256$, $d=1$, $s=2$
conv3	$k=[3,3,3]$, $c=128$, $d=4$, $s=1$	$k=[3,3,3]$, $c=512$, $d=1$, $s=2$
conv4	$k=[3,3,3]$, $c=128$, $d=8$, $s=2$	$k=[3,3,3]$, $c=512$, $d=1$, $s=2$
final_conv	$k=[1,1,1]$, $c=128$, $d=1$, $s=1$	$k=[1,1,1]$, $c=512$, $d=1$, $s=1$

Table 6: Sparse Convolutional Network Parameters of the *RISE*-2 Policy. Both the sparse encoder and the spatial aligner utilize MinkResNet [20] for point cloud encoding. k, c, d, s stand for the kernel size, output channel number, dilation, and stride in the convolutional layers, respectively.

Training. We train the ***RISE*-2** policy on 4 NVIDIA A100 GPUs for 60k steps with a batch size of 240, an initial learning rate of 3×10^{-4}, and a warmup step of 2000. We employ a cosine scheduler to adjust the learning rate during training. During training, 20% of the color images are augmented using color jitter with (brightness, contrast, saturation, hue) parameters set to (0.4, 0.4, 0.2, 0.1).

Inference. We deploy ***RISE*-2** using an RTX 2060 SUPER GPU and achieve a 5Hz inference speed, demonstrating efficient performance compared to larger vision-language-action policies [7]. We additionally provide scripts that enable policy inference to be executed on a remote server.

B.3 Visualization of Sparse Semantic Features

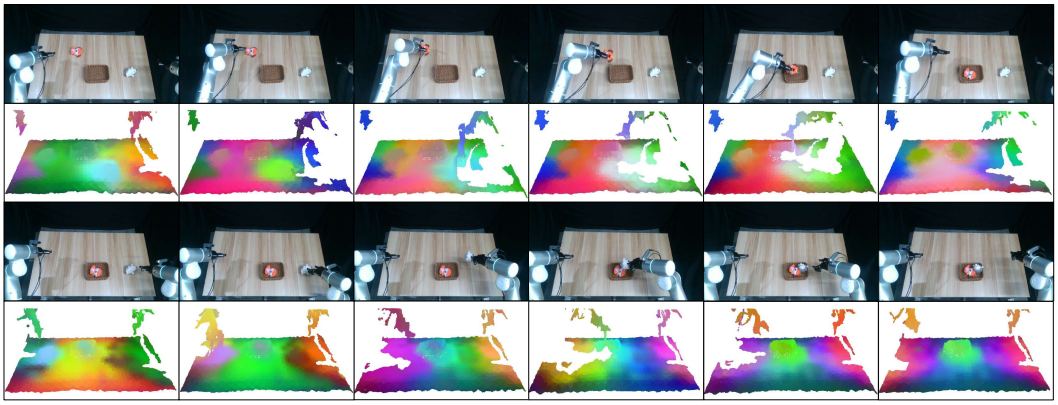


Figure 10: Visualization of Sparse Semantic Features. The colors are obtained by performing PCA on the features. The original sparse semantic features are aligned to the input point cloud using weighted spatial interpolation function in the spatial aligner for clearer visualization.

Fig. 10 visualizes the sparse semantic features obtained from the dense encoder by projecting the 2D feature map to 3D form using the reference coordinates. The sparse semantic features are aligned to the input point cloud using weighted spatial interpolation, as detailed in §3.2. Although the 2D

feature map output by the dense encoder is in low resolution (32×18 for DINOv2 [75] backbone), we still observe clear and distinguishable continuous feature variations on the aligned features, where the targets at the current step can be easily identified from the entire scene. Such a characteristic ensures precise feature fusion in the spatial domain. Additionally, we find that the features from the visual foundation model DINOv2 [75] change significantly as the task progresses, enabling the model to clearly understand the global state at the current time.

C Experiments Details

C.1 Setup

Data Collection. The teleoperated demonstrations are collected using AirExo [29], while in-the-wild demonstrations are collected and adapted using our proposed ***AirExo*-2** system. For each task, unless further specified, we collect 50 teleoperated demonstrations for ***RISE*-2** policy evaluation and 50 in-the-wild demonstrations for ***AirExo*-2** system evaluations.

Baselines. We compare ***RISE*-2** against several representative policies based on 2D images and 3D point clouds, including: (1) *ACT* [123], which employs transformers to map image observations and proprioception to robot action chunks; (2) *Diffusion Policy* [18], which formulates action prediction as a diffusion denoising process [37, 92] conditioned on the image observations; (3) *CAGE* [111], an extension of Diffusion Policy that incorporates visual foundation models [75], a causal observation perceiver [44], and an attention-based diffusion action head for improved generalization; (4) *RISE* [102], a 3D imitation policy that leverages a sparse 3D encoder for efficient point cloud perception; and (5) π_0 [7], a vision-language-action flow model for general robot control.

Evaluation Protocols. All policies except π_0 are deployed on a workstation with an NVIDIA RTX 2060 SUPER GPU, while π_0 is deployed remotely on a server with an NVIDIA A100 GPU for GPU memory considerations. Following the procedure outlined in [19, 111], we adopt a consistent evaluation method for each policy to minimize performance variation and ensure reproducibility. Specifically, we generate uniformly distributed test positions randomly before each task evaluation. The workspace is set up identically across different policies and test environments, and success rates are recorded for each test case. Each policy is evaluated over 20 consecutive trials per task (except for the ***Serve Steak*** task, where we evaluate each policy for 10 consecutive trials), and the success rates are computed accordingly.

Generalization Evaluation. We select the ***Collect Toys*** task to conduct a generalization experiment to evaluate the robustness of different policies under varying levels of environmental disturbances. The policies are trained with demonstrations from a narrow domain, with only one training background and one target object. Then, we introduce two types of disturbances during generalization evaluations: background variations and object differences, as illustrated in Fig. 11. We then follow the same evaluation protocol introduced above to compute the policy success rates under (1) novel backgrounds, (2) novel objects, and (3) both novel backgrounds and novel objects, enabling a comprehensive comparison of each policy's generalization capability. For evaluation concerning the background variations, we use each of the 5 novel backgrounds for 4 evaluation trials in the same order to ensure fairness.

Figure 11: Generalization Evaluation Setup. We evaluate policy generalization using 5 unseen backgrounds and 1 unseen target object. Training is done in a narrow domain with only one background and one target object.

C.2 Action Accuracy Analysis Details

To evaluate the action accuracy of the data collection systems, we designed a special evaluation board with three tracks, as illustrated in Fig. 12. Each track has fixed holes spaced 2 cm apart. We created custom connectors for both the ***AirExo*-2** and UMI [19] that fit into these fixed holes, allowing us to collect position data. By sequentially placing the connectors into each fixed hole

along the track, we can calculate the relative movement distance between two adjacent fixed holes and compare it with the true value (20 mm) to calculate the error.

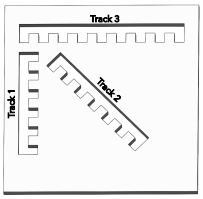
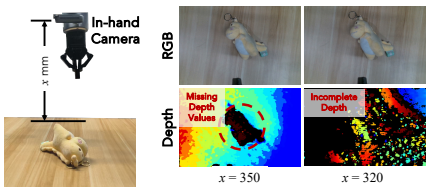


Figure 12: The Evaluation Board.

C.3 Case Study: Are In-Hand Cameras Sufficient for Manipulations?

We conducted a case study to investigate whether in-hand cameras are sufficient for many manipulation tasks. We select the ***Collect Toys*** task as an example and utilize CAGE [111] as the policy for this case study.

In-hand cameras alone are often insufficient for manipulation tasks and may pose additional obstacles on policy learning. As shown in Fig. 13 (bottom), neither global nor in-hand cameras alone provide adequate observations for 2D image-based policies to achieve strong performance. Recent work [102] has demonstrated that using only a global camera enables a 3D imitation policy to outperform 2D multi-view image-based policies, highlighting the importance of 3D information for scene understanding. However, as illustrated in Fig. 13 (top), in-hand cameras may produce incomplete depth information when the robotic arm approaches an object, making them unsuitable for 3D point-cloud-based policies. Consequently, relying solely on in-hand cameras can degrade the performance of the policies, particularly for 3D policies that rely on complete and accurate depth information to achieve superior performance.



Method	# Cameras	Success Rate
global only	1	45.0%
in-hand only	2	60.0%
global + in-hand	3	72.5%

Figure 13: Case Study Results. (Top) In-hand cameras often yield low-quality depth during object interaction, limiting their use for 3D point cloud-based policies. (Bottom) This case study uses CAGE with relative actions, as in the original implementation [111]; subsequent experiments use CAGE with absolute actions.

C.4 Additional Experiment Results

Learning from Pseudo-Robot Demonstrations. We also evaluate the performance of the RISE policy [102] using pseudo-robot demonstrations. Specifically, we employ ***AirExo-2*** to collect an equivalent number of in-the-wild demonstrations, which are then transformed into pseudo-robot data and used to train the RISE policy. The trained policy is deployed on a real robot *without any fine-tuning*. As shown in Fig. 14 and Tab. 7, RISE achieves good zero-shot performance, though with a slightly larger drop compared to ***RISE-2***. These findings highlight the critical role of a generalizable policy in successfully transferring manipulation skills from in-the-wild demonstrations to real robots, under the circumstances of no teleoperation data.

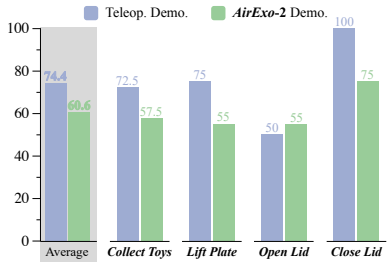


Figure 14: RISE Policy Performance with Different Demonstrations. Generalizable policies are crucial for learning from in-the-wild demonstrations.

Data Source	Policy	Average Success Rate (4 tasks) ↑
Teleoperation	RISE [102]	74.375%
	RISE-2	**93.750%**
AirExo-2	RISE [102]	60.625% $^{-13.750\%}$
	RISE-2	**85.000%** $^{-8.750\%}$

Table 7: Evaluation Results of Different Policies in Learning from Pseudo-Robot Demonstrations. ***RISE-2*** shows a smaller performance drop than RISE [102] when trained with adapted pseudo-robot demonstrations.

Action Representations. We conduct additional experiments on action representations for the ***Collect Toys*** task, comparing relative and absolute action representations. The results in Tab. 8 show that while relative action representation sometimes yields better results, absolute action representation provides more stable performance, particularly in terms of generalization. Therefore, we use absolute action representations for Diffusion Policy [18] and CAGE [111] throughout our experiments, except for the case study in Appendix C.3. For π_0 [7], we still use relative action representation for consistency with its released pre-trained models.

Method	Action Representation	Success Rate ↑		
		in-domain	background	object
Diffusion Policy [18]	relative	37.5%	**20.0%**	**5.0%**
	absolute	**40.0%**	12.5%	**5.0%**
CAGE [111]	relative	**72.5%**	32.5%	35.0%
	absolute	65.0%	**45.0%**	**42.5%**

Table 8: Evaluation Results of Different Action Representations on the *Collect Toys* Task. Generally, absolute action representation leads to a more stable performance than relative action representation.

C.5 Failure Analysis

Imprecise Action. During the policy rollouts, we observe that most failures stem from imprecise predicted actions. For example, in the ***Open Lid*** task, the lid protrudes only about 1.5 cm from the coffee machine body, requiring highly accurate end-effector control from the visuomotor policies. At the same time, the left gripper must stabilize the machine to prevent it from shifting on the table while the right arm applies force to lift the lid. Baseline failures typically arise from inaccuracy in either arm — where the left arm fails to stabilize the coffee machine or the right arm fails to open the lid. In contrast, our ***RISE*-2** policy produces more precise actions, resulting in fewer failures, even when trained with adapted in-the-wild data.

Risk of Visual Overfitting to Robot States. We also observe that in bimanual manipulation tasks, since both arms occupy a large portion of the observation image, *there is an increased risk of visuomotor policies overfitting to robot states*. This occurs even without proprioceptive inputs, as the policy may implicitly infer actions from the visual configuration of the arms. The issue is further compounded by limited training data: with only 50 demonstrations, most baselines fail to learn sufficiently accurate actions, leading to low phase success rates. Such overfitting also reduces the likelihood of performing recovery behaviors after phase failures, which becomes especially evident in long-horizon and complex tasks such as ***Serve Steak***, ultimately causing frequent task failures. In contrast, ***RISE*-2** produces more precise actions, resulting in higher success rates. Nevertheless, all policies still exhibit varying degrees of overfitting to robot visual state and struggle to recover from failures in the absence of recovery demonstrations, highlighting an important direction for developing more generalizable robotic visuomotor policies.

D Author Contributions

The author contributions for this paper are as follows. For each item, authors are listed in order of contribution, and <u>underlined names</u> indicate equal contributions.

- **Project Lead**: Hongjie Fang.
- **System Design and Implementation**.
  - ***AirExo*-2 Hardware**: Yiming Wang, Hongjie Fang, Hao-Shu Fang.
  - ***AirExo*-2 Visual Adaptors**: Jingjing Chen, Hongjie Fang.
  - ***RISE*-2 Policy**: Chenxi Wang, Hongjie Fang, Jingjing Chen, Shangning Xia.
- **Experiments**:
  - **Design**: <u>Hongjie Fang</u>, <u>Chenxi Wang</u>, <u>Jingjing Chen</u>, Yiming Wang, Hao-Shu Fang.
  - **Platform Setup**: Hongjie Fang, Yiming Wang.
  - **Data Collection**: <u>Hongjie Fang</u>, <u>Chenxi Wang</u>, <u>Jingjing Chen</u>.

- **Data Processing**: Jingjing Chen, Zihao He, Xiyan Yi, Yunhan Guo.
- **Baselines**: Hongjie Fang, Shangning Xia, Chenxi Wang, Jingjing Chen.
- **Policy Training**: Hongjie Fang, Chenxi Wang, Shangning Xia, Jingjing Chen, Xinyu Zhan, Lixin Yang.
- **Policy Evaluation**: Hongjie Fang, Chenxi Wang, Jingjing Chen.
- **User Study**: Hongjie Fang, Chenxi Wang, Jingjing Chen, Xiyan Yi.
- **Accuracy Analysis**: Hongjie Fang, Jun Lv, Jingjing Chen, Yiming Wang.
- **Scalability Analysis**: Hongjie Fang, Jingjing Chen, Chenxi Wang, Hao-Shu Fang.

- **Writing**: Hongjie Fang, Chenxi Wang, Jingjing Chen, Yiming Wang, Hao-Shu Fang.
- **Project Advisor**: Hao-Shu Fang, Cewu Lu, Weiming Wang.