

H2R: A Human-to-Robot Data Augmentation for Robot Pre-training from Videos

Guangrun Li^{1*}, Yaoxu Lyu^{1*}, Zhuoyang Liu^{1*}, Chengkai Hou^{1†}, Jieyu Zhang², Shanghang Zhang¹ 

¹State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University;

²University of Washington

* Equal contribution, † Project lead,  Corresponding author

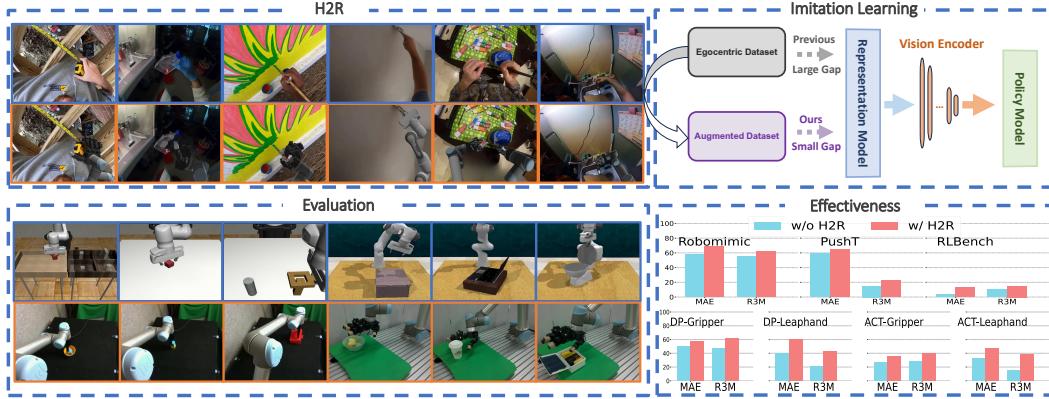


Figure 1: **Overview of H2R.** H2R is a data augmentation technique designed to enhance robot pre-training by converting first-person human hand operation videos into robot-centric visual data. By bridging the visual domain gap, H2R improves pre-trained visual encoders for downstream robot policies (imitation learning), validated across simulation benchmarks and real-world robotic tasks.

Abstract

Large-scale pre-training using videos has proven effective for robot learning. However, the models pre-trained on such data can be suboptimal for robot learning due to the significant visual gap between human hands and those of different robots. To remedy this, we propose **H2R**, a simple data augmentation technique that detects human hand keypoints, synthesizes robot motions in simulation, and composites rendered robots into egocentric videos. This process explicitly bridges the visual gap between human and robot embodiments during pre-training. We apply H2R to augment large-scale egocentric human video datasets such as Ego4D and SSv2, replacing human hands with simulated robotic arms to generate robot-centric training data. Based on this, we construct and release a family of 1M-scale datasets covering multiple robot embodiments (UR5 with gripper/Leaphand, Franka) and data sources (SSv2, Ego4D). To verify the effectiveness of the augmentation pipeline, we introduce a CLIP-based image-text similarity metric that quantitatively evaluates the semantic fidelity of robot-rendered frames to the original human actions. We validate H2R across three simulation benchmarks—Robomimic, RL Bench, and PushT and real-world manipulation tasks with a UR5 robot equipped with Gripper and Leaphand end-effectors. H2R consistently improves downstream success rates, yielding gains of **5.0%–10.2%** in simulation and **6.7%–23.3%** in real-world tasks across various visual encoders and policy learning methods. These results indicate that H2R improves the generalization ability of robotic policies by mitigating the visual discrepancies between human and robot domains.

1 Introduction

Pre-training of generalizable robotic features for object manipulation and motion navigation constitutes a crucial objective within the realm of robotics. Inspired by the remarkable accomplishments of large scale pre-training in computer vision [1, 2, 3, 4, 5] and natural language processing [6, 7, 8, 9, 10], many efforts have been devoted to harness large-scale data to construct generalizable representations in the robotics field [11, 12, 13]. Nevertheless, when it comes to robot manipulation, the process of collecting demonstrations is labor-intensive and expensive [14, 15, 12, 16, 17, 18, 19, 20, 21, 22]; meanwhile, there exist many large-scale egocentric video datasets showing how humans perform manipulation and navigation, which can potentially serve as a cheap alternative of demonstrations for the pre-training of generalizable visual features for robotics.

Recent works [2, 23, 24] analyze such egocentric human video datasets such as Ego4D [25], SSv2 [26], and Epic Kitchens [27] with the aim of gleaning prior knowledge about object manipulation and enabling the acquisition of general and robust feature representations. However, during the representation learning, the gap in visual representations between the human arm and the robotic arm remains largely unaddressed and can hinder the transferability of models trained on egocentric datasets to robotic systems. Specifically, when utilizing the robot expert data to fine-tune the pre-trained robotic representations for downstream robotic tasks, the model has to learn to bridge the visual gap between the first-person human hand and the robots in addition to acquiring nuanced task-specific skills demonstrated in the robot expert data. This would result in increased complexity during the fine-tuning process and suboptimal performance.

To mitigate this issue, we propose H2R (as shown in Figure 1), a simple data augmentation method that converts videos of **Human** hand operations into that of **Robotic** arm manipulation. H2R consists of two major procedures: the first part is to generate the robotic movements to imitate the human hand movements in a video, followed by the second stage that overlays the robotic movements onto the human hand’s movements in the video. Specifically, in the **first** part, we employ state-of-the-art 3D hand reconstruction model HaMeR [28] to accurately detect the position and posture of the human hand in egocentric videos. Then, we simulate the same robot state in simulators to obtain the mask of robot actions. In the **second** stage, we use the Segment Anything Model [29] to automatically separate human hand from background, and use the inpainting model LaMa [30] to fill the removed hand mask. After that, we align the camera intrinsic parameters of the images detected in HaMeR with those in the simulator, and then achieve pixel-level matching between the robotic arm images in the simulators and the human hand images in the egocentric video. Finally, we overlay the robotic arm images captured by the simulator’s camera onto the areas where the human hands are removed. Through such a process, H2R explicitly reduces the gap between human and robot hands by creating realistic robotic arm movements that visually mimic human hand actions. It allows the model to learn the task-specific actions demonstrated by the human hand, but with robotic arm visual representations that are more suitable for robotic systems.

Based on this pipeline, we construct and release a series of large-scale robot-centric datasets (**H2R-1M**), each containing approximately 1 million augmented frames. These datasets cover multiple robot embodiments (UR5 with gripper/Leaphand, Franka) and human video sources (SSv2, Ego4D), and provide modular annotations to support downstream robotic learning. To evaluate the effectiveness of the H2R augmentation process, we introduce a CLIP-based semantic similarity metric that measures how well the rendered robot frames preserve the original action semantics. This provides a lightweight and scalable proxy to assess the alignment quality between input human videos and robot-augmented outputs.

To further verify the utility of the augmented datasets, we conduct downstream experiments comparing models pre-trained on original egocentric datasets with those trained on our released H2R-enhanced datasets. These comparisons are performed on both simulation and real-world robotic manipulation tasks to assess how H2R impacts policy learning performance in practice. For the pre-training stage, we apply MAE [1] and R3M [2] frameworks to train visual encoders using 62,500 videos from the SSv2 dataset, where 16 keyframes are sampled per video. The pretrained visual representations are used in downstream imitation learning pipelines. We conduct comprehensive evaluations across both simulation and real-world manipulation tasks. In simulation, we evaluate pre-trained visual encoders on three benchmark suites: Robomimic, RLBench, and PushT. H2R improves the average success rates by +10.2% for MAE and +6.3% for R3M on Robomimic tasks, +10.0% for MAE and +5.0% for R3M on RLBench tasks, and +5.3% for MAE and +7.0% for R3M on the PushT task. In real-

world experiments, we deploy H2R-enhanced encoders on a UR5 robot equipped with both Gripper and Leaphand end-effectors, evaluated over six manipulation tasks under two policy frameworks: Diffusion Policy (DP) [31] and ACT [32]. H2R brings consistent improvements, with average success rate gains of +13.3% (MAE) and +17.0% (R3M) under DP, and +10.8% (MAE) and +17.5% (R3M) under ACT. H2R also improves MAE/R3M performance when pre-trained on Ego4D, with up to +10% and +15% gains in simulated and real-world tasks, respectively. These results demonstrate that H2R effectively bridges the human-to-robot visual domain gap and significantly enhances downstream performance across both simulated and real-world robotic manipulation settings.

Our paper provides three contributions:

- We propose a data-centric pipeline, H2R, to mitigate the gap between human and robot hands when utilizing large-scale egocentric video datasets to pre-train generalizable visual features for robots.
- We construct and release diverse robot-centric datasets (H2R-1M), combining multiple embodiments and egocentric video sources to support robot-compatible visual pretraining.
- We demonstrate the effectiveness of H2R through extensive experiments on closed-loop benchmarks.

2 Related Work

Robot Imitation Learning. Data-driven policy learning [33, 34, 23, 35, 31, 11] has enabled robots to autonomously perform tasks such as grasping, locomotion, and manipulation. Imitation learning [31, 36, 32, 37] trains policies from successful demonstrations, often supervised by behavior cloning [38, 39] objectives. ACT [32] addresses non-Markovian dynamics by fusing temporal sequences, while diffusion models [31, 36] are introduced to handle the inherent multimodality of robot motions. In addition, CordViP [40] leverages 3D object-robot correspondences to enhance dexterous manipulation.

Visual Encoder Pretraining for Robotics. Visual pretraining improves generalization of robotic policies across diverse tasks. Researchers have explored architectural designs [41, 42], training objectives [43, 44, 45], and dataset compositions [46, 47, 48, 49]. PVR-Control [50] shows that pretrained visual representations can outperform direct state-based policies. RPT [51] tokenizes observations to enable masked prediction pretraining. Methods like MVP [52] and R3M [2] utilize self-supervised objectives on videos to learn representations transferable to reinforcement learning. Voltron [4] demonstrates the use of MAE and contrastive learning for hierarchical robot control.

Cross-Domain Visual Alignment. Bridging the domain gap between human and robot visual inputs remains a major challenge. WHIRL [53] matches task structure from third-person views, while RoVi-Aug [54] and Mirage [55] manipulate appearance via segmentation or image-space preprocessing. EgoMimic [56] removes hands and normalizes views to align egocentric perspectives.

3 H2R: Human-to-Robot Data Augmentation

In this section, we describe **H2R**, a data augmentation pipeline for robot learning from egocentric human videos (Figure 2). It replaces human hands in every frame with robotic arms equipped with various end effectors, generating a new, visually different dataset. This approach aims to mitigate the visual gap between human hands and robots, thereby improving the generalizability of visual representations learned from egocentric data to robotic domains. Figure 3 illustrates examples of human hand videos that have been processed using H2R. These examples include different types of robots (UR5 and Franka) equipped with a variety of end-effectors, such as dexterous hands and grippers. We further introduce a CLIP-based metric to evaluate semantic consistency, and construct a family of large-scale datasets that combine different robot embodiments (UR5 with gripper, Leaphand, or both; Franka with gripper) and egocentric sources (SSv2 and Ego4D).

3.1 H2R Data Augmentation Pipeline

3D Hand Pose Estimation. In order to overlay the human hands in the egocentric image with different robots, we first need an efficient and accurate model to detect hand information. We adopt HaMeR [28], a state-of-the-art model for 3D hand detection and reconstruction, to accurately locate

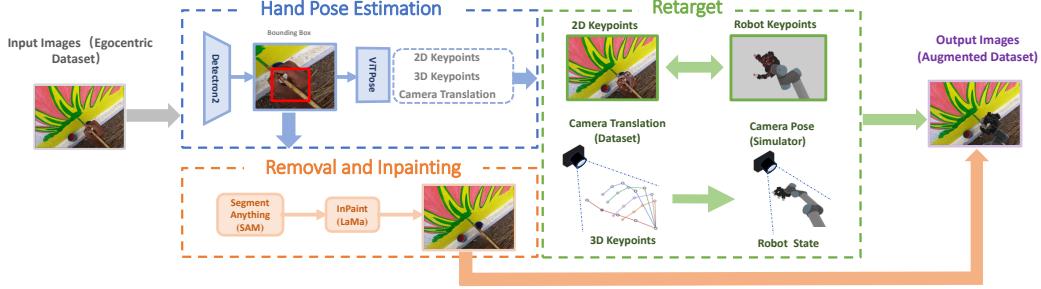


Figure 2: H2R Pipeline. H2R involves replacing human hands with robotic arms by first using the HaMeR model to detect hand poses and camera parameters. The human hand is then removed using the SAM, and the inpainting model LaMa fills in the gap. A robot hand is constructed based on the detected pose and keypoints, with the camera perspective adjusted to match the original image. Finally, the robot hand is overlaid onto the image, ensuring accurate alignment with the human hand.

the hand in the image, providing precise positional information for subsequent hand removal. Given an ego-centric RGB image, HaMeR estimates both the hand pose (including 3D keypoints) and the intrinsic and extrinsic parameters of the rendering camera.

Human Arm and Hand Removal. After obtaining the hand pose information, we need to mask out the human hand and then inpaint the masked area with a proper background. We use Segment Anything Model (SAM) [29] to automatically segment the human hand and arm regions using the hand pose information detected by HaMeR. Then, to obtain clean backgrounds for inserting robotic arms, we apply LaMa [30], a powerful inpainting model, to fill in the removed hand-arm region. This yields clean RGB images without human limbs, providing a seamless background for inserting robotic arms in the subsequent steps.

robotic arm and End Effector Construction. This step involves constructing the robotic arm and different end effectors. There are two common types of end-effectors in robotic manipulation: grippers and dexterous hands. In our pipeline, we handle them differently based on their structural characteristics. For dexterous hands, which have multiple degrees of freedom and resemble human hands, we compute the joint angles by analyzing the hand keypoints predicted by HaMeR. Specifically, each finger joint angle is determined by the angle formed between three consecutive keypoints corresponding to that finger segment. For grippers, which typically consist of two or more parallel fingers, we determine the degree of opening or closing based on the Euclidean distance between the relevant fingertips in the hand keypoints. However, since hand keypoints alone are insufficient to fully define the robotic arm’s complete configuration, especially for joints that do not directly correspond to hand movement, we manually assign plausible values to these undetermined joint positions to construct a reasonable arm pose.

Simulator Camera Position Alignment. The visual bias introduced by the camera perspective is more significant than the action retargeting itself. To address this, we use the hand keypoints and camera parameters from HaMeR to adjust the camera pose in the simulator. Specifically, we define two coordinate systems: C_H , the coordinate system aligned with the human hand, and C_S , the coordinate system of the robotic arm in the simulator. By mapping the position of the camera in C_H to C_S , we can ensure that the camera in the simulator shares the same perspective as the one captured in the real-world egocentric human image. The original camera position ${}^W \text{cam}_{\text{real}}$ in the world frame is transformed to the aligned simulator position ${}^W \text{cam}_{\text{sim}}$ using transformations from human hand (${}_H \mathbf{R}$) and robot simulator (${}_S \mathbf{R}$) coordinate systems:

$${}^W \text{cam}_{\text{sim}} = {}_S^W \mathbf{R} \times {}_H^W \mathbf{R}^{-1} \times {}^W \text{cam}_{\text{Real}} \quad (1)$$

A more detailed explanation of the coordinate transformation and camera alignment process is provided in Appendix B.

Robot Hand Rendering and Copy-paste. After setting the camera, the segmentation mask of the robotic arm is obtained by shooting with the camera. We directly obtain the pixel coordinates of the human hand keypoints from HaMeR, which are predicted from the input image. In parallel, the pixel coordinates of the robot end-effector links are computed in the simulator by projecting their 3D positions through the aligned camera using the known transformation matrices. By aligning the

robot link positions with the corresponding human hand keypoints in pixel space, we ensure that the overlaid robot hand accurately matches the position and orientation of the original hand in the image, achieving precise pixel-level alignment.

3.2 Data Quality Evaluation

To assess the visual plausibility and semantic consistency of the robot-augmented images produced by H2R, we employ a vision-language similarity evaluation based on CLIP [57]. This method measures how well the rendered robot actions align with high-level semantic descriptions of manipulation tasks. For each augmented frame, we formulate a pair of textual prompts describing the same action: one from a human-centric perspective and one from a robot-centric perspective. The human-centric template is defined as: A human is *[action]*”; the robot-centric template is defined as: A robotic arm is *[action]*” . Here, *[action]* is a natural language phrase describing the high-level behavior, such as holding a bottle” or “brushing paint on a wall.”

We use the CLIP ViT-B/32 model to compute cosine similarity between the image embedding of the augmented frame and the corresponding textual prompt. This yields a scalar score reflecting how semantically consistent the image content is with the robot-centric description. Similarly, we compute the similarity between the original human frame and the human-centric prompt. This setup allows us to directly compare semantic alignment before and after augmentation under their respective modalities. For each frame, only the action-relevant region is retained, and background context is preserved via inpainting using LaMa. We average similarity scores over multiple samples per action class to mitigate variance and isolate the impact of robotic replacement. All textual prompts are manually curated to maintain consistency across frames and embodiments.

This evaluation protocol provides a lightweight, scalable, and interpretable metric for measuring the effectiveness of visual hand-to-robot transformations in preserving task semantics. Detailed results and further analyses are presented in Appendix D.

3.3 H2R Dataset Construction

We construct four large-scale robot-centric video datasets using the proposed H2R augmentation pipeline, each combining a specific robot embodiment with a human egocentric video dataset. These datasets are designed to support policy learning, visual representation learning, and embodiment generalization.

Applying H2R to SSv2. We begin with the SSv2 dataset, which contains 220,847 video clips of human actions with everyday objects, designed to help models understand fine-grained hand gestures. From the official training split, we select 62,500 videos that cover a wide variety of manipulation categories. For each video, we uniformly sample 16 keyframes, resulting in exactly 1,000,000 frames. Each sampled frame is passed through the H2R pipeline to replace the human hand with a simulated robotic arm. For the UR5 robot, we construct three distinct datasets with different end-effector configurations: **H2R-UR5-SSv2-1M-Gripper** uses a standard two-finger parallel gripper; **H2R-UR5-SSv2-1M-Leaphand** employs a four-finger anthropomorphic Leaphand; and **H2R-UR5-SSv2-1M-Mix** combines both embodiments at the frame level to increase visual diversity and generalization. For the Franka robot, which is configured only with a two-finger gripper, we construct **H2R-Franka-SSv2-1M** following the same augmentation strategy.

Applying H2R to Ego4D. To expand the applicability of our dataset to unconstrained, real-world human activity, we apply the same augmentation pipeline to the **Ego4D** dataset, which features long-form, first-person videos of natural daily behavior across diverse environments. Following the action-centric preprocessing strategy introduced by MPI [3], we extract 117,624 action clips from 2,486 Ego4D videos, each consisting of three keyframes. These are further processed to produce **H2R-UR5-Ego4D-1M-Gripper**, **H2R-UR5-Ego4D-1M-Leaphand**, and **H2R-UR5-Ego4D-1M-Mix**, using the same embodiment configuration logic as in the SSv2 case. For Franka, we similarly construct **H2R-Franka-Ego4D-1M** using the gripper configuration.

Each dataset contains approximately one million augmented images. In addition to the RGB images, we provide comprehensive frame-level metadata to support downstream use. Each entry includes: (1) the original label (from SSv2 or Ego4D), (2) the 3D hand keypoints, camera intrinsics, and extrinsics estimated by HaMeR, (3) the inpainted background image generated by LaMa, and (4) the final robot



Figure 3: **Examples of H2R Augmentation.** Each column shows images before and after augmentation. The top row uses UR5 Leaphand, the middle row uses UR5 Gripper, and the bottom row uses Franka to replace human hands.

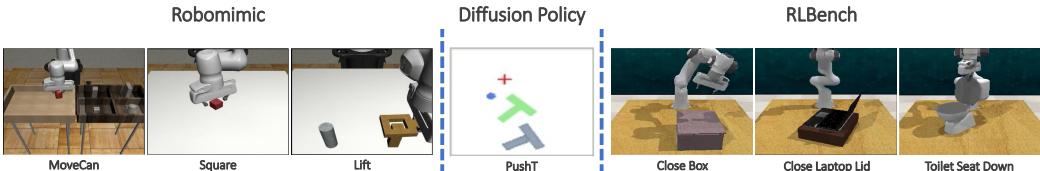


Figure 4: **Simulation Benchmark Overview.** Visualization of the seven simulation tasks used for evaluation, including three from Robomimic (MoveCan, Square, Lift), one PushT task from Diffusion Policy, and three from RLbench (Close Box, Close Laptop Lid, Toilet Seat Down).

joint configurations and camera pose alignment used in the simulator for rendering. All robotic arms are rendered using the SAPIEN simulator, with camera viewpoint precisely matched to the original egocentric perspective using a two-step transformation between the HaMeR-estimated camera pose and the robot-world coordinate frame. Each dataset is released in a modular file structure, with paired directories for images, masks, and JSON metadata files. This organization ensures flexible access for both visual pre-training pipelines and embodied policy learning frameworks.

4 Simulation Experiment

4.1 Experiment Setup

Encoder Pre-training. We adopt the MAE [1, 23] and R3M [2] frameworks for pre-training, each employing a Vision Transformer (ViT) Base [58] model as the visual encoder. The training dataset used is either **Ego4D** (117K clips) or **H2R-UR5-SSv2-1M-Mix** (1M H2R-augmented images). Both MAE and R3M are pre-trained with $8 \times$ A800 GPUs. MAE employs 800 epochs with 128 batch size and 4e-4 learning rate, while R3M uses 20K steps with 256 batch size and 1e-4 learning rate. For policy training, we present the set of parameters that necessitate policy learning in Appendix C.

Simulation Benchmark. For each pre-training method, we evaluate the performance of pre-trained encoders in imitation learning. Specifically, we select a total of seven simulation tasks in different environments, which are from Robomimic [37], RLbench [59], and Diffusion Policy [31]. In particular, for Robomimic, we train the policies using the behavior cloning (BC) and evaluate them on tasks such as **MoveCan**, **Square**, and **Lift**, where the robot performs actions such as moving or lifting objects. For RLbench, we train the policies with Diffusion Policy and evaluate them on three manipulation tasks: **Close Box**, **Close Laptop Lid** and **Toilet Seat Down**. We also use the **PushT**

Table 1: Simulation Benchmark Results. Success rates (%) across imitation learning tasks in Robomimic, PushT, and RLbench environments, evaluated with MAE and R3M encoders before and after applying H2R. **Blue** represents an increase in task success rate, while **Red** represents a decrease. All subsequent tables follow the same rule.

	Robomimic				Diffusion Policy	RLBench			
	MoveCan	Square	Lift	Average		PushT	CloseBox	CloseLaptopLid	ToiletSeatDown
MAE (SSv2)	54	25.5	94.5	58	59.2	0	10	0	3.3
MAE (H2R-SSv2)	79.5 (+25.5%)	29.5 (+4.0%)	95.5 (+1.0%)	68.2 (+10.2%)	64.5 (+5.3%)	5 (+5%)	15 (+5.0%)	20 (+20.0%)	13.3 (+10%)
R3M (SSv2)	59.5	20.5	85	55	15	0	20	10	10
R3M (H2R-SSv2)	61.5 (+2.0%)	37.5 (+17.0%)	85.0 (0.0%)	61.3 (+6.3%)	22.0 (+7.0%)	5 (+5.0%)	20 (0.0%)	20.0 (+10.0%)	15.0 (+5.0%)

task in the Diffusion Policy evaluation framework, which evaluates a robot’s ability to push an object to a target location. An overview of these simulation tasks is shown in Figure 4.

4.2 Simulation Results

Table 1 shows that the encoders trained on **H2R-UR5-SSv2-1M-Mix** dataset significantly outperform those trained on the original **Ego4D** data across all simulation tasks. Specifically, for Robomimic tasks, the MAE encoder achieves an average success rate improvement of 10.2%, while the R3M encoder improves by 6.3%. In particular, the MoveCan task exhibits a substantial **25.5%** increase for MAE trained on the H2R-1M dataset. For the PushT task, H2R leads to success rate gains of 5.3% (MAE) and 7.0% (R3M), further confirming the generalization benefits brought by our method. Moreover, in the RLbench benchmark, H2R consistently enhances performance, achieving an average improvement of 10.0% for MAE and 5.0% for R3M.

These results demonstrate that H2R-1M dataset delivers superior visual representations for imitation learning compared to large-scale human video datasets like Ego4D. H2R achieve an improvement in the pre-training performance of the vision encoder on imitation learning by bridging the visual gap between human hands and robotic arms.

4.3 Performance on Other Datasets

In addition to the **H2R-UR5-SSv2-1M-Mix** dataset, we also perform the same experiments on the PushT task and the tasks in RLbench benchmark using the **H2R-UR5-Ego4D-1M-Mix** dataset. The experimental results in the simulator are shown in Table 2, which denotes that H2R remains highly effective when applied to the Ego4D dataset.

Table 2: Results on Ego4D Dataset. Success rates (%) across imitation learning tasks in the PushT and RLbench environments.

	Diffusion Policy	RLBench			
		PushT	CloseBox	CloseLaptopLid	ToiletSeatDown
MAE (Ego4D)	51.3	0	0	5	1.7
MAE (H2R-Ego4D)	53.5 (+2.2%)	10 (+10%)	5 (+5.0%)	0 (-5.0%)	5 (+3.3%)
R3M (Ego4D)	13.6	10	5	5	6.7
R3M (H2R-Ego4D)	13.5 (-0.1%)	15 (+5.0%)	5 (0.0%)	15.0 (+10.0%)	11.7 (+10.0%)

5 Real World Experiment

Table 3: Real-world Task Results. We report the success rate (%) over real-world tasks for MAE and R3M.

Policy	Task	MAE (SSv2)	MAE (H2R-SSv2)	R3M (SSv2)	R3M (H2R-SSv2)	Policy	Task	MAE (SSv2)	MAE (H2R-SSv2)	R3M (SSv2)	R3M (H2R-SSv2)
DP	Gripper-PickCube	45	65 (+20%)	40	50 (+10%)	ACT	Gripper-PickCube	25	30 (+5%)	25	30 (+5%)
	Gripper-Stack	50	55 (+5%)	55	70 (+15%)		Gripper-Stack	20	35 (+15%)	20	40 (+20%)
	Gripper-CloseBox	55	50 (-5%)	45	65 (+20%)		Gripper-CloseBox	35	40 (+5%)	40	50 (+10%)
	Average	50	56.7 (+6.7%)	46.7	61.7 (+15%)		Average	26.7	35 (+8.3%)	28.3	40 (+11.7%)
	LeapHand-GraspChicken	40	55 (+15%)	10	35 (+25%)		LeapHand-GraspChicken	45	50 (+5%)	10	35 (+25%)
	LeapHand-StandCup	35	60 (+25%)	20	50 (+30%)		LeapHand-StandCup	25	50 (+25%)	20	60 (+40%)
LeapHand	LeapHand-OpenBox	45	65 (+20%)	40	45 (+5%)	GRIPPER	LeapHand-OpenBox	30	40 (+10%)	15	20 (+5%)
	Average	40	60 (+20%)	21.7	43.3 (+21.7%)		Average	33.3	46.7 (+13.3%)	15	38.3 (+23.3%)

5.1 Experiment setup

Robot Setups. We validate the effectiveness of H2R in real-world manipulation tasks using a UR5 robotic arm with two different end effectors: Gripper [60] and Leaphand [61]. Realsense [62] is mounted on the side of the robotic arm, which provides a similar viewpoint to the ego-centric human video data used in the pre-trained visual model. Real-world setups are shown in Figure 6.

Real-world Tasks. We set up six tasks for gripper manipulation and dexterous manipulation (Figure 5). For the Gripper tasks, we design: (1) **Gripper-PickCube**, where the Gripper picks up a

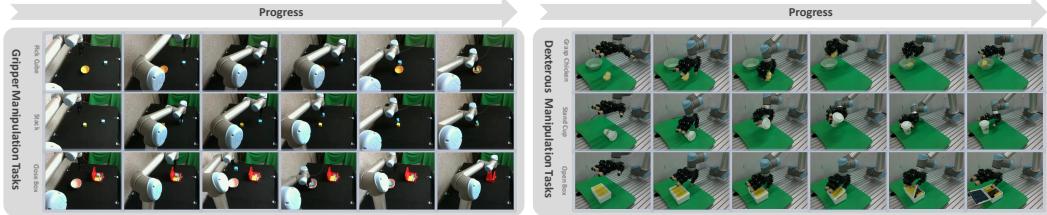


Figure 5: **Visualization of Real-world Manipulation Tasks.** The left columns show Gripper tasks and the right columns show Leaphand tasks. Each task is illustrated with six frames, demonstrating the progression from the initial state to the completion of the manipulation.

cube and places it into a bowl; (2) **Gripper-Stack**, where a blue cube is stacked atop a yellow cube; and (3) **Gripper-CloseBox**, where a cube is retrieved from a box, placed into a bowl, and the box lid is subsequently closed. For the Leaphand tasks, we design: (4) **Leaphand-GraspChicken**, where a toy chicken is grasped and placed into a bowl; (5) **Leaphand-StandCup**, where a fallen cup is stood upright on the table; and (6) **Leaphand-OpenBox**, where an articulated box lid is opened.

Data Collection. We collect expert demonstrations through human teleoperation. For gripper manipulation, we use keyboard-based teleoperation. For dexterous manipulation, we teleoperate Leaphand using a vision-based retargeting system. Due to differences in task complexity, robot embodiment, and teleoperation methods, we adopt varying numbers of demonstrations, episode lengths per demonstration, and maximum action steps during evaluation for each task, as in Table 4.

Policy Training. For policy training, we select the Diffusion Policy (DP) [31] and ACT [32] as policy frameworks. We apply the pre-trained MAE and R3M visual encoders to downstream policy learning, following the same pretraining configuration described in Section 4.1. DP and ACT are trained for 300 epochs for each gripper manipulation task and 3000 epochs for each dexterous manipulation task.

Evaluation. To ensure that the initial conditions are consistent throughout the evaluation, we randomly place the target objects within a predefined area following a uniform distribution during expert demonstrations. During the evaluation, each real-world task is rolled out **20** times. We report the success rates of these tasks to assess model quality.

5.2 Real-world Results

In the real-world tasks, we employ diffusion policy (DP) [31] and ACT [32] to drive robots in performing real-world tasks, where the visual encoders are pre-trained using MAE and R3M. Policy training details are presented in Appendix C.

To evaluate the effectiveness and generalization of H2R, we conduct real-world experiments under two different robot embodiments: UR5 (same embodiment as downstream execution) / Franka (cross-embodiment generalization).

To assess the effectiveness of H2R, we use **H2R-UR5-SSv2-1M-Mix** during pre-training, which matches the embodiment used in policy training and evaluation. As shown in Table 3, H2R provides consistent improvements across both MAE and R3M encoders for both DP and ACT policies. For Leaphand tasks under DP, H2R leads to an average improvement of **20%** (MAE) and **21.7%** (R3M), while for Gripper tasks, improvements reach **6.7%** (MAE) and **15%** (R3M). Similar trends are observed under ACT, with Leaphand performance increasing by **13.3%** (MAE) and **23.3%** (R3M), and Gripper tasks improving by **8.3%** and **11.7%**, respectively.

Table 4: **Task-specific Data Collection and Evaluation Settings.**

Task	Episode Length	Num Demos	Max Steps
Gripper-PickCube	45	30	80
Gripper-Stack	43	30	80
Gripper-CloseBox	68	30	120
Leaphand-GraspChicken	150	50	500
Leaphand-StandCup	150	50	500
Leaphand-OpenBox	200	50	800

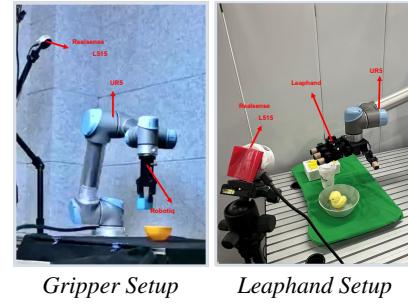


Figure 6: **Real-world Robot Setups.** Experimental setups using a UR5 robot with two different end effectors: a parallel-jaw gripper (left) and a dexterous Leaphand hand (right).

Table 6: Real-world Success Rates with H2R. We report the task success rates (%) using two robot embodiments (UR5, Franka) for data augmentation during pre-training. All models are evaluated on downstream tasks using the Leaphand end-effector, under MAE and R3M encoders with DP and ACT policies. Despite embodiment mismatch, H2R provides consistent gains over the baseline without augmentation.

Policy	Task	MAE (SSv2)	MAE (H2R-UR5-SSv2)	MAE (H2R-Franka-SSv2)	R3M(SSv2)	R3M (H2R-UR5-SSv2)	R3M (H2R-Franka-SSv2)
DP	Leaphand-GraspChicken	40	55 (+15%)	35 (-5%)	10	35 (+25%)	20 (+10%)
	Leaphand-StandCup	35	60 (+25%)	50 (+15%)	20	50 (+30%)	30 (+10%)
	Leaphand-OpenBox	45	65 (+20%)	45	40	45 (+5%)	45 (+5%)
	Average	40	60 (+20%)	43.3 (+3.3%)	21.7	43.3 (+21.7%)	31.7 (+10%)
ACT	Leaphand-GraspChicken	45	50 (+5%)	50 (+5%)	10	35 (+25%)	40 (+30%)
	Leaphand-StandCup	25	50 (+25%)	50 (+25%)	20	60 (+40%)	25 (+5%)
	Leaphand-OpenBox	30	40 (+10%)	30	15	20 (+5%)	5 (-10%)
	Average	33.3	46.7 (+13.3%)	43.3 (+10%)	15	38.3 (+23.3%)	23.3 (+8.3%)

To evaluate cross-embodiment generalization, we use **H2R-Franka-SSv2-1M** dataset. Downstream policy training and evaluation are still conducted with UR5 Leaphand. As shown in Table 6, H2R still outperforms raw Ego4D despite the embodiment mismatch. Under DP, MAE improves from **40%** to **43.3%**, and R3M from **21.7%** to **31.7%**. Under ACT, MAE increases from **33.3%** to **43.3%**, and R3M from **15%** to **23.3%**. Although Franka-based pretraining underperforms UR5, the results confirm H2R’s robustness to embodiment variations. Additional analysis, including failure case categorization and experiments on generalization under lighting perturbations, is provided in Appendix E and G.

5.3 Performance on Other Datasets

We compare encoders pre-trained on the original **Ego4D** dataset and on the H2R-augmented **H2R-UR5-Ego4D-1M-Mix**, and apply them to real-world Leaphand manipulation tasks using the ACT [32] policy. As shown in Table 5, models trained with H2R data consistently outperform those using raw Ego4D across all three Leaphand manipulation tasks, achieving an average improvement of **15%** for MAE and **6.7%** for R3M. These results complement our findings on SSv2 and reinforce that robot-centric augmentation improves downstream real-world performance.

5.4 Ablation study

To evaluate the effectiveness of each component in H2R, we conduct ablation studies on two time-consuming steps: (1) performing hand inpainting without overlaying a robotic arm (H2R w/o Overlay), and (2) overlaying the arm without precise alignment between the hand and the camera, instead using random pasting (H2R w/o Retarget). Table 7 shows the necessity and effectiveness of each component in H2R. The first step leads to a significant drop in success rate due to the loss of critical human-object interaction pixels after inpainting. The second step fails to provide accurate motion cues for the model and introduces visual mismatches with real-world manipulation tasks.

6 Conclusion

We propose H2R, a data augmentation technique that bridges the visual gap between human hand demonstrations and robotic arm manipulations by replacing human hands in first-person videos with robotic arm movements. Using 3D hand reconstruction and image inpainting models, H2R generates synthetic robotic arm manipulation sequences, making them more suitable for robot pre-training. Experiments across simulation benchmarks and real-world tasks demonstrate consistent improvements in success rates for encoders trained with various pre-training methods (e.g., MAE, R3M), highlighting the effectiveness and generalization of H2R. H2R enables efficient transfer of task knowledge from human demonstrations to robotic systems, reducing the reliance on costly robot-specific data collection. Finally, we expressed the broader impact of this work in Appendix H.

Table 5: Real-world Success Rates with Ego4D Pre-training. We report success rates (%) on real-world Leaphand tasks using visual encoders pre-trained on the Ego4D dataset. All downstream policies are trained using ACT.

Task	MAE (Ego4D)	MAE (H2R-Ego4D)	R3M (Ego4D)	R3M (H2R-Ego4D)
Leaphand-GraspChicken	25	35 (+10%)	15	20 (+5%)
Leaphand-StandCup	25	50 (+25%)	20	25 (+5%)
Leaphand-OpenBox	35	45 (+10%)	30	40 (+10%)
Average	28.3	43.3 (+15%)	21.7	28.3 (+6.7%)

Table 7: Ablation Study. Ablation by removing robot overlay (w/o Overlay) and camera-hand retargeting (w/o Retarget). Task success rates (%) are reported under DP and ACT policies. Removing either component causes significant performance degradation, with red values showing the relative drop compared to full H2R.

Policy	Task	H2R	H2R w/o Overlay	H2R w/o Retarget
DP	Leaphand-GraspChicken	55	30 (-25%)	30 (-25%)
	Leaphand-StandCup	60	40 (-20%)	55 (-5%)
	Leaphand-OpenBox	65	20 (-45%)	45 (-20%)
	Average	60	30 (-30%)	43.3 (-16.7%)
ACT	Leaphand-GraspChicken	50	25 (-25%)	45 (-5%)
	Leaphand-StandCup	50	35 (-15%)	30 (-20%)
	Leaphand-OpenBox	40	25 (-15%)	15 (-25%)
	Average	46.7	28.3 (-18.3%)	30 (-16.7%)

References

- [1] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [2] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.
- [3] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, Heming Cui, Bin Zhao, Xuelong Li, Yu Qiao, and Hongyang Li. Learning manipulation by predicting interaction, 2024.
- [4] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023.
- [5] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [7] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022.
- [8] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [9] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.
- [10] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.
- [11] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.
- [12] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale, 2023.
- [13] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.
- [14] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. Ar2-d2: Training a robot without a robot. In *Conference on Robot Learning*, pages 2838–2848. PMLR, 2023.
- [15] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning, 2022.
- [16] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset, 2024.
- [17] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot, 2023.
- [18] Jensen Gao, Annie Xie, Ted Xiao, Chelsea Finn, and Dorsa Sadigh. Efficient data collection for robotic manipulation via compositional generalization, 2024.

- [19] Alex X Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes, 2021.
- [20] Alexander Herzog, Kanishka Rao, Karol Hausman, Yao Lu, Paul Wohlhart, Mengyuan Yan, Jessica Lin, Montserrat Gonzalez Arenas, Ted Xiao, Daniel Kappler, et al. Deep rl at scale: Sorting waste in office buildings with a fleet of mobile manipulators, 2023.
- [21] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale, 2021.
- [22] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*, 2024.
- [23] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.
- [24] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024.
- [25] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
- [26] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.
- [27] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.
- [28] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers, 2023.
- [29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.
- [30] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions, 2021.
- [31] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.
- [32] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [33] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.
- [34] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning, 2024.
- [35] Jingyun Yang, Zi ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim(3)-equivariant diffusion policy for generalizable and data efficient learning, 2024.
- [36] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

- [37] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation, 2021.
- [38] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.
- [39] Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning, 2021.
- [40] Yankai Fu, Qiuxuan Feng, Ning Chen, Zichen Zhou, Mengzhen Liu, Mingdong Wu, Tianxing Chen, Shanyu Rong, Jiaming Liu, Hao Dong, et al. Cordvip: Correspondence-based visuomotor policy for dexterous manipulation in real-world. *arXiv preprint arXiv:2502.08449*, 2025.
- [41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [43] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.
- [44] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.
- [45] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.
- [46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.
- [48] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8429–8438, 2019.
- [49] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021.
- [50] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17359–17371. PMLR, 17–23 Jul 2022.
- [51] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, pages 683–693. PMLR, 2023.
- [52] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training, 2022.
- [53] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild, 2022.
- [54] Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning, 2024.

- [55] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *7th Annual Conference on Robot Learning*, 2023.
- [56] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024.
- [57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision, 2021.
- [58] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- [59] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.
- [60] Robotiq Inc. Adaptive grippers. <https://robotiq.com/products/adaptive-grippers>, 2025. Accessed: 2025-02-01.
- [61] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning, 2023.
- [62] Intel Corporation. Intel®realsense™lidar camera l515. <https://www.intelrealsense.com/lidar-camera-l515/>, 2025. Accessed: 2025-02-01.

A Limitations

H2R is a simple data augmentation technique for robot pre-training from videos. There are several limitations, which open up possibilities for future work:

1. H2R augmentation pipeline is sensitive to the accuracy of pose estimation and hand segmentation, which demands high-quality egocentric video datasets as prerequisites. This sensitivity becomes particularly pronounced in challenging scenarios involving low-light conditions or partial hand occlusions, where current estimation methods often fail to maintain required precision.
2. While our robotic platform incorporates diverse manipulator configurations spanning various robotic arm, grippers and dexterous hands, the current implementation remains limited to single-arm operation. Future work will expand this framework to dual-arm and humanoid robotics, significantly broadening the model’s application scope.
3. It would be valuable to extend our work on more tasks besides manipulation, like visual navigation and mobile manipulation.

B Details of Simulator Camera Position Alignment

We define two coordinate systems: C_H , the coordinate system aligned with the human hand, and C_S , the coordinate system of the robot arm in the simulator. We build the coordinate system ${}^W\mathbf{I}_H$ based on the hand keypoints:

$${}^W\mathbf{I}_H = \{{}^w\mathbf{i}_{H,x}, {}^w\mathbf{i}_{H,y}, {}^w\mathbf{i}_{H,z}\} \quad (2)$$

Where ${}^w\mathbf{i}_{H,x}, {}^w\mathbf{i}_{H,y}, {}^w\mathbf{i}_{H,z}$ are unit vectors along the x-axes, y-axes and z-axes of the human hand coordinate system. With the keypoints get in HaMeR, we build the three axis of coordinates with the following functions:

$$\begin{aligned} {}^w\mathbf{i}_{H,x} &= {}^w\mathbf{i}_{0,9} \\ {}^w\mathbf{i}_{H,y} &= {}^w\mathbf{i}_{0,9} \times {}^w\mathbf{i}_{0,13} \\ {}^w\mathbf{i}_{H,z} &= {}^w\mathbf{i}_{H,x} \times {}^w\mathbf{i}_{H,y} \end{aligned} \quad (3)$$

Where ${}^w\mathbf{i}_{0,9}$ and ${}^w\mathbf{i}_{0,13}$ are unit vectors along the middle and ring fingers, respectively. In this notation, the first index (0) refers to the specific finger (middle or ring), and the second index (9 and 13) corresponds to the joint numbers along those fingers, as defined by the MANO model. Similarly, To construct the mapping from hand pose to robot arms, we need to get another coordinate system ${}^W\mathbf{I}_S$ in the simulator:

$${}^W\mathbf{I}_S = \{{}^w\mathbf{i}_{S,x}, {}^w\mathbf{i}_{S,y}, {}^w\mathbf{i}_{S,z}\} \quad (4)$$

The method of determining the axis of coordinates is the same:

$$\begin{aligned} {}^w\mathbf{i}_{S,x} &= {}^w\mathbf{i}_{0,2} \\ {}^w\mathbf{i}_{S,y} &= {}^w\mathbf{i}_{0,2} \times {}^w\mathbf{i}_{0,3} \\ {}^w\mathbf{i}_{S,z} &= {}^w\mathbf{i}_{S,x} \times {}^w\mathbf{i}_{S,y} \end{aligned} \quad (5)$$

Where $\mathbf{i}_{0,2}, \mathbf{i}_{0,3}$ are unit vectors along robot fingers that correspond to human middle and ring fingers and the index corresponds to the joint numbers defined by MANO. We build the following two coordinate transformation matrix to construct the mapping:

$$\begin{aligned} {}_H^W\mathbf{R} &= \begin{pmatrix} {}^W\mathbf{I}_H & \mathbf{key}_{human} \\ \mathbf{O} & 1 \end{pmatrix} \\ {}_S^W\mathbf{R} &= \begin{pmatrix} {}^W\mathbf{I}_S & \mathbf{key}_{robot} \\ \mathbf{O} & 1 \end{pmatrix} \end{aligned} \quad (6)$$

Where $\mathbf{key}_{human}, \mathbf{key}_{robot}$ are the positions of human wrist and robot wrist. After obtaining the two coordinate systems, we need to determine the position of the camera in the simulator (${}^W\mathbf{cam}_{sim}$)

and the position of the camera in the real world (${}^H \mathbf{cam}_{Real}$), thus we can ensure we get the same pose of the human hand and robot arms

$$\begin{aligned} {}^H \mathbf{cam}_{Real} &= {}^W_H \mathbf{R}^{-1} \times {}^W \mathbf{cam}_{Real} \\ {}^S \mathbf{cam}_{sim} &= {}^H \mathbf{cam}_{Real} \\ {}^W \mathbf{cam}_{sim} &= {}^W_S \mathbf{R} \times {}^H_W \mathbf{R}^{-1} \times {}^W \mathbf{cam}_{Real} \end{aligned} \quad (7)$$



Figure 7: **H2R samples.** Visual comparison between original human data (top) and our augmented data (bottom).

C Policy Training Details

Policy Training in Simulation Experiment. For Robomimic tasks, we train for 200 steps and report the mean success rate. For tasks in RLBench, we train for 800 epochs for each policy model and test them 20 times in the RLBench environment with a random initialization. For the PushT task, we train the Diffusion Policy model for 200 epoches and report the success rate in the simulation environment. The training hyperparameters used in this work are identical to those described in the original paper.

Policy Training in Real-world Experiment. We select the Diffusion Policy (DP) [31] and ACT [32] as policy frameworks. We apply the pre-trained MAE and R3M visual encoders to downstream policy learning, following the same pretraining configuration described in Section 4.1. DP and ACT are trained for 300 epochs for each gripper manipulation task and 3000 epochs for each dexterous manipulation task.

D Evaluation of H2R Effectiveness

In Figure 7 are six pairs of subfigures, each showing a human action and its corresponding augmented robotic action. The six action pairs in Figure 7 (left to right) are described as follows:

- holding a yellow measuring tape at a construction site
- holding a bottle
- painting on the wall with a brush / Brushing paint on a wall
- turning on the kitchen faucet
- preparing painting materials on a table

The similarity scores computed by CLIP are presented in Table 8. Across these samples, the augmented images consistently exhibit higher similarity scores with the robot-centric action descriptions compared to the similarity scores between the unaugmented images and the human-centric action descriptions. This quantitative analysis demonstrates that the H2R augmentation effectively enhances the visual alignment between human and robotic actions.

E Failure Case Analysis

To better understand the limitations of our policy and the challenges encountered in real-world deployments, we present a qualitative analysis of failure cases from two representative tasks: a

Table 8: **CLIP similarity scores.** Higher values indicate better alignment between images and action descriptions.

	Img1	Img2	Img3	Img4	Img5	Img6
ori	30.6	23.7	31.2	31.7	27.5	28.7
aug	32.6	28.9	32.3	32.0	28.7	31.0

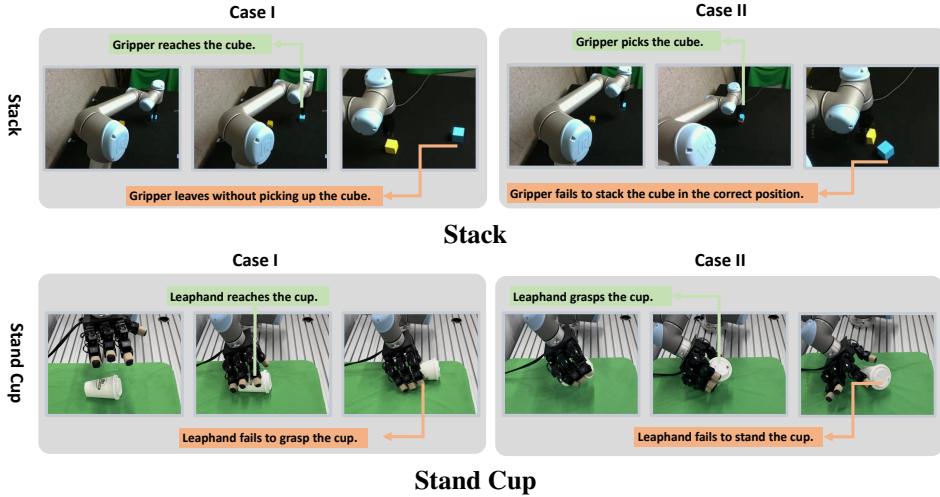


Figure 8: **Failure case visualizations: Stack and Stand Cup.** We visualize real-world manipulation executions for two downstream tasks: *Stack* (top) and *Stand Cup* (bottom). These images provide qualitative insights into the performance and failure modes of the policy in real deployment, highlighting challenges such as object misalignment, perception noise, and grasp precision.

Table 9: **Sub-goal.Task-specific sub-goal evaluation.** To gain fine-grained insights into policy performance, we design a manual rubric covering key sub-goals for each manipulation task. Each cell reports the number of successful vs. unsuccessful attempts (Y/N) over 20 evaluation trials. Results show that models enhanced with H2R consistently accomplish more sub-goals across tasks compared to their baseline counterparts, demonstrating improved robustness in real-world execution. Bold numbers indicate better performance between paired models.

Task	Sub-goal	MAE(Y/N)	MAE+H2R(Y/N)	R3M(Y/N)	R3M+H2R(Y/N)
Gripper-PickCube	Overall success?	9/11	13/7	8/12	10/10
	Pick up the cube?	14/6	15/5	11/9	13/7
Gripper-Stack	Overall success?	10/10	11/9	11/9	14/6
	Pick up the cube?	13/7	16/4	13/7	17/3
Gripper-CloseBox	Overall success?	11/9	10/10	9/11	13/7
	Place the cube in the bow?	12/8	14/6	12/8	15/5
Leaphand-GraspChicken	Overall success?	8/12	11/9	2/18	7/13
	Pick up the chicken?	13/7	14/6	3/17	10/10
Leaphand-StandCup	Overall success?	7/13	12/8	4/16	10/10
	Pick up the cup?	12/8	18/2	12/8	15/5
Leaphand-OpenBox	Overall success?	9/11	13/7	8/12	9/11
	Identify contact location?	14/6	16/4	10/10	10/10

gripper-based task (*Gripper-Stack*) and a dexterous manipulation task (*Gripper-StandCup*). Figure 8 illustrates typical failure modes observed during execution.

In the **Gripper-Stack** task, we identify two major failure scenarios:

Case I: Grasp Failure Unnoticed. The robot arm fails to successfully grasp the blue cube. However, the policy proceeds as if the object had been grasped, moving toward the yellow cube and attempting to perform the stacking operation. This leads to a complete task failure.

Case II: Misaligned Placement. The robot successfully grasps the blue cube but fails to align it correctly on top of the yellow cube during the stacking phase, resulting in an unstable or failed placement.

In the **Stand Cup** task, similar issues emerge due to perception and control limitations:

Case I: Grasp Position Error. The Leaphand end-effector attempts to grasp the cup but fails to target the correct contact region. As a result, the cup slips out of the grasp during lifting, preventing task completion.

Case II: Insufficient Lifting Trajectory. Even when the grasp is successful, the lifting motion lacks sufficient amplitude or stability to fully stand the cup upright. The cup either tips over or fails to stand securely.

To enable fine-grained evaluation of policy performance and gain deeper insights into failure cases, we designed a task-specific evaluation rubric. Table 9 displays our rubric that the evaluator filled out when rolling out different policies. Take the DP policy as an example, the results in Table 9 demonstrate that H2R-augmented visual representation models not only improve overall success rates in real-world tasks, but also allow to accomplish more than half of the task consistently.

Table 10: **Robomimic Experiment result.** We report the success rate (%) over IL-based tasks for MAE and R3M Robomimic.

	MoveCan	Square	Lift	Average		PushT
MAE	54	25.5	94.5	58		59.2
MAE+CutMix1	72.0 (+18.0%)	30.0 (+4.5%)	95.0 (+0.5%)	65.7 (+7.7%)		37.5 (-21.7%)
MAE+CutMix2	58.0 (+4.0%)	36.0 (+10.5%)	90.0 (-4.5%)	61.3 (+3.3%)		40.0 (-19.2%)
MAE+CutMix3	78.0 (+24.0%)	32.0 (+9.3%)	92.0 (-2.5%)	67.3 (+2.7%)		42.0 (-17.2%)
MAE+H2R	79.5 (+25.5%)	29.5 (+4.0%)	95.5 (+1.0%)	68.2 (+10.2%)		64.5 (+5.3%)
R3M	59.5	20.5	85	55		15
R3M+CutMix1	69.5 (+10.0%)	30.0 (+9.5%)	91.0 (+6.0%)	63.5 (+8.5%)		19.0 (+4.0%)
R3M+CutMix2	66.0 (+6.5%)	26.0 (+5.5%)	83.0 (-2.0%)	58.3 (+3.3%)		17.0 (+2.0%)
R3M+CutMix3	68.0 (+8.5%)	26.0 (+5.5%)	84.0 (-1.0%)	59.3 (+4.3%)		14.0 (-1.0%)
R3M+H2R	61.5 (+2.0%)	37.5 (+17.0%)	85.0 (0.0%)	61.3 (+6.3%)		22.0 (+7.0%)

F Additional Ablation Study in Simulation Experiment

In addition to pre-training on the H2R data and raw data, we also applied a simple CutMix baseline to demonstrate the effectiveness of using the robotic arm to cover the human hand, which overlays a fixed set of specific images of robotic arms with grippers onto the original images, ensuring that the overlaid images cover the human hands as much as possible, without exceeding the detected bounding box. Our H2R is different from such baseline by employing robot hand construction to better match the pose of the hand and arm in the images. Based on the type of robotic arm used in CutMix, we categorize the augmented set into three types: CutMix1 represents the UR5 robotic arm, CutMix2 refers to the Franka robotic arm, and CutMix3 combines both the UR5 and Franka robotic arms.

From Table 10, we observe that the encoder trained on H2R processed data shows consistent improvements across various tasks compared to the encoder trained on the original data, with the average success rate on all tasks ranging from 0.9% to 10.2%. Especially for the more challenging MoveCan task, it can improve the success rate by 25.5%. Additionally, while encoders trained on the relatively simple CutMix data show improvement on tasks in Robomimic, their performance

in the PushT task remains slightly worse than the encoders trained on original data. These results demonstrate the effectiveness of using the robotic arm to cover the human hand in video data, as well as the effectiveness of H2R in imitation learning.

Table 11: Generalization under Lighting Variations. Success rates (%) under real-world lighting disturbances. Models trained with additional lighting augmentations (LightAug) show significant improvements in robustness compared to baseline and H2R-only models.

Tasks	MAE	MAE+H2R	MAE+H2R+LightAug	R3M	R3M+H2R	R3M+H2R+LightAug
Leaphand-GraspChicken	10	10	20	0	0	10
Leaphand-StandCup	20	25	40	5	15	20
Leaphand-OpenBox	5	0	25	10	10	10
Average	11.7	11.7	28.3	5	8.3	13.3

G Generalization on Light Condition.

To evaluate the generalization under varying lighting conditions, we introduce illumination disturbances during evaluation, as illustrated in Figure 9. Additionally, during training, we incorporate randomized lighting with varying directions and colors into the simulation environment for data augmentation. We compare three settings: no augmentation(MAE, R3M), H2R augmentation(MAE+H2R, R3M+H2R), and H2R with lighting disturbances(MAE+H2R+LightAug, R3M+H2R+LightAug). As shown in Table 11, the model trained with H2R and lighting perturbations demonstrates significantly better generalization to real-world lighting variations than other baselines, highlighting the effectiveness of H2R in bridging the domain gap caused by lighting variations.

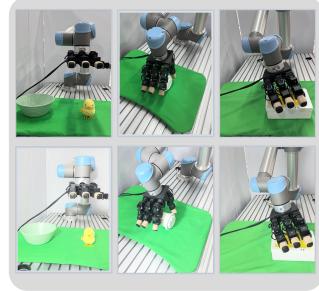


Figure 9: Lighting Setup in Real-World Experiments. Light configuration used in real-world evaluations.

H Broader Impact

In this work, we proposed a simple data augmentation technique that detects human hand keypoints, synthesizes robot motions in simulation, and composites rendered robots into egocentric video. We focused on the theme above and doesn't have direct social impact. We hope that through our data augmentation methods, more egocentric human datasets can be used for higher-level robot manipulation models, thereby promoting progress in the field of embodied intelligence.