MotionTrans: Human VR Data Enable Motion-Level Learning for Robotic Policies

Chengbo Yuan[1,2], **Rui Zhou***[5], **Mengzhen Liu***[3], **Yingdong Hu**[1,2], **Shengjie Wang**[1,2], **Li Yi**[1,2], **Chuan Wen**[4], **Shanghang Zhang**[3], **Yang Gao**[1,2]†

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]Shanghai Qi Zhi Institute [3]School of Computer Science, Peking University
[4]Shanghai Jiao Tong University [5]Wuhan University
* Equal contribution † Corresponding author

Abstract: Scaling real robot data is a key bottleneck in imitation learning, leading to the use of auxiliary data for policy training. While other aspects of robotic manipulation such as image or language understanding may be learned from internet-based datasets, acquiring motion knowledge remains challenging. Human data, with its rich diversity of manipulation behaviors, offers a valuable resource for this purpose. While previous works show that using human data can bring benefits, such as improving robustness and training efficiency, it remains unclear whether it can realize its greatest advantage: *enabling robot policies to directly learn new motions for task completion*. In this paper, we systematically explore this potential through multi-task human-robot cotraining. We introduce *MotionTrans*, a framework that includes a data collection system, a human data transformation pipeline, and a weighted cotraining strategy. By cotraining 30 human-robot tasks simultaneously, we direcly transfer more than 10 motions from human data to deployable end-to-end robot policies. Notably, 9 tasks achieve non-trivial success rates in zero-shot manner. *MotionTrans* also significantly enhances pretraining-finetuning performance (+40% success rate). Through ablation study, we also identify key factors for successful motion learning: cotraining with robot data. These findings unlock the potential of motion-level learning from human data, offering insights into its effective use for training robotic manipulation policies. All data, code, and model weights will be open-sourced.

Keywords: human data, motion transfer, cotraining, policy learning.

1 Introduction

Learning robotic manipulation policies from teleoperated demonstrations has progressed rapidly in recent years [1, 2, 3]. However, collecting large-scale robot datasets remains costly and labor-intensive [4, 5], creating a significant bottleneck for further improvement of manipulation abilities. To address data scarcity, researchers have turned to auxiliary sources, such as images or language [6, 7] to help policy training. While internet data provides abundant vision-language knowledge to aid policy learning [8], acquiring motion knowledge remains a significant challenge.

Human data [9, 10] represents a particularly promising source to solve this: it is abundant, easy to collect, and rich in diverse manipulation behaviors [10]. Previous works have leveraged human demonstrations to extract task-aware representations, such as affordances [11] or keypoint flows [12], to support motion transfer. However, the introduction of intermediate representation hinders integration with mainstream end-to-end policies. More recently, with advances in wearable sensing, researchers begin to explore the use of human motion data (with hand poses recorded from VR device) directly for robot policy cotraining or pretraining [9, 13, 14, 15, 16]. These approaches have shown benefits for visual grounding [15], robustness [14] and training efficiency [16]. However, it is

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

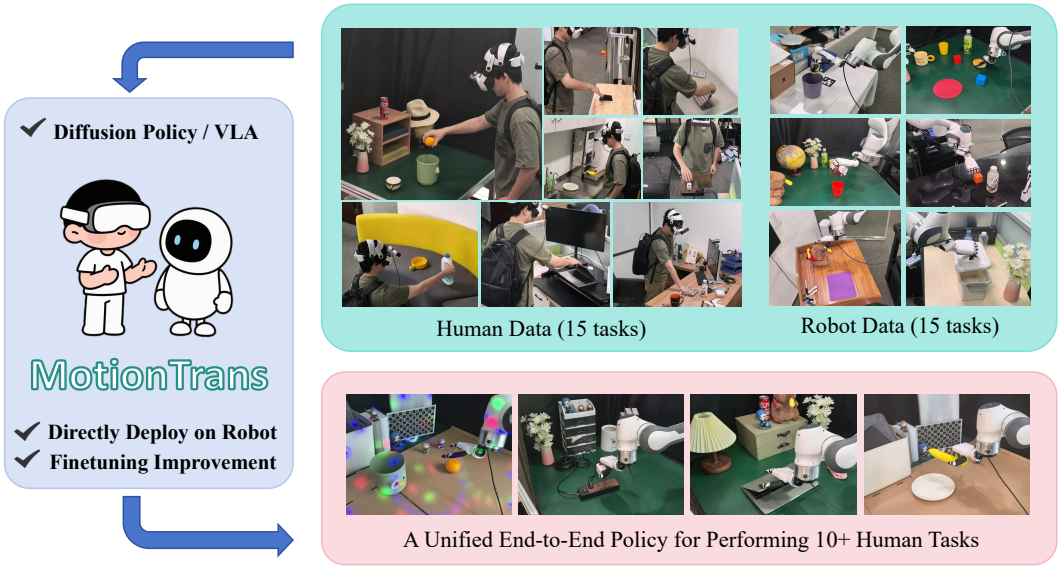


Figure 1: We propose *MotionTrans*, a framework that enables **motion-level** learning from VR human data. By cotraining on 15 human tasks and 15 robot tasks, we empower end-to-end robotic manipulation policies to directly perform tasks in human data on real robot. Finetuning performance is also improved when a few robot demonstrations are available for these tasks.

still uncertain whether it can fully realize its greatest advantage: *allowing robot policies to directly acquire new motions for task completion*.

In this paper, we investigate this question by introducing *MotionTrans* (Figure 1), a framework designed to **directly learn 10+ robot-executable motions from human data for a unified, end-to-end robot policy.** This is achieved through multi-task human-robot cotraining. We develop a VR-based teleoperation system and data collection pipeline to construct the *MotionTrans Dataset*, which includes 3,213 demonstrations across 15 human tasks and 15 robot tasks from more than 10 scenes. We further propose a transformation procedure that maps human demonstrations into the robot's observation–action space, making them compatible with mainstream end-to-end policies such as Diffusion Policy [2] or the Vision-Language-Action model (π_0-VLA) [3]. Finally, we adopt a weighted cotraining strategy that jointly optimizes over both human and robot tasks. We name our framework *MotionTrans* because it enables motion transfer from human data to deployable robot policies.

We first evaluate the zero-shot performance on all human tasks. This means that we directly deploy policies to robot without collecting any robot data for these human tasks. Results show that Diffusion Policy [2] and π_0-VLA model [3] achieve non-trivial success rates for 9 tasks in total. Even in unsuccessful cases, they exhibit meaningful motion for task completion, such as reaching target objects. We also find that, when few robot demonstrations of these human tasks are available for finetuning, pretraining on the *MotionTrans Dataset* leads to an average 40% boost in success rate on these tasks. Further analysis indicates that the effectiveness of motion transfer depends on the presence of robot data for human-robot cotraining. Together, these findings highlight the possibility for motion-level learning from human data, and provide a clear framework and principles for achieving this. Our contributions can be summarized as:

- *MotionTrans* framework for end-to-end human-to-robot motion transfer, including data collection system, *MotionTrans Dataset*, a pipeline to transform human data into robot format, and a weighted human-robot cotraining strategy.
- ***MotionTrans* enables explicit human motions transfer for end-to-end robot policies, even for zero-shot settings** (directly learn more than 10 tasks from human data to robot). It also improve finetuning performance with +40% success rate on average.

2 Related Work

Imitation Learning for Robot Manipulation. Imitation learning [17, 18] has made significant progress in recent years. By learning motion from training data [1, 19], imitation policies can effectively perform a wide range of manipulation tasks [2, 20], including challenging multi-task settings [3, 16, 21, 22]. In this paper, we focus on two widely-used architectures for imitation learning: Diffusion Policy [2] and the π_0 Vision-Language-Action Model (π_0-VLA) [3]. However, the scalability of training data remains a major challenge, due to the high cost of collecting real-robot data [4, 5]. This has led to the use of auxiliary data [6, 7] for policy training. Despite ability such as image or language understanding in robotic manipulation could improve from internet-based pretraining [8, 23], acquiring motion knowledge remains difficult. Human data [10, 24, 25, 26], with its abundant and diverse manipulation behaviors, provides a valuable supplement for this.

Task-Aware Representation Learning from Human. Early works have leveraged task-aware representations for human-to-robot knowledge transfer. Self-supervised learning has been used for implicit task-aware representations [27, 28, 29, 30, 31] learning, while representations like affordances [11, 32, 33], object poses [34], videos [35, 36], and motion flows [12, 37, 38, 39] support motion-aware representation learning. Some approaches use wrist trajectories as prompts for one-shot human-to-robot skill transfer [40, 41, 42, 43, 44]. EgoZero [45] predicts wrist poses from smart glasses, but relies on keypoint-based representations [46] for policy observations. The use of intermediate representations in these methods limits their integration with mainstream end-to-end visuomotor policy learning [2, 3], restricting their future applicability.

End-to-End Policy Learning with Posed Human Data. Recent advancements in wearable sensing [9, 19, 47] now allow easy collection of posed human data (with hand keypoints, wrist poses information etc.) through VR devices [10]. This data provide action label for prediction, supporting end-to-end policy learning [48]. Some studies cotrain human and robot data [9, 13, 49, 50, 51], while others first pretrain with human data and then finetune with robot demonstrations [14, 15, 16]. These works have shown policy improvements in visual grounding [15], robustness [13, 14], and training efficiency [16]. However, whether it can achieve direct transfer of motions from human to robot remains unclear [45]. To the best of our knowledge, our paper is the first to systematically verify motion-level end-to-end learning from human data.

3 MotionTrans

In this section, we present our proposed *MotionTrans* framework (Figure 2). We first introduce the motion transfer problem and define the observation-action space of the policy (Section 3.1). To facilitate human-robot data cotraining, we develop data collection systems for both human and robot data (Section 3.2). We then propose a pipeline to convert human data into a robot format (Section 3.3). This ensures compatibility with mainstream robot policies, enabling subsequent end-to-end cotraining. Finally, we choose the architecture of robot policies and apply human-robot multi-task cotraining (Section 3.4).

3.1 Problem Definition

Our goal is to enable motion-level policy learning from human data through human-robot multi-task cotraining. Specifically, we aim to train a policy P_{policy} on the combined dataset $D = D_{\text{robot}} \cup D_{\text{human}}$, where $D_{\text{robot}} = \{D^i_{\text{robot}} \mid i = 1, \ldots, N_{\text{robot}}\}$ is the robot dataset, and $D_{\text{human}} = \{D^i_{\text{human}} \mid i = 1, \ldots, N_{\text{human}}\}$ is the human dataset. Each D^i represents a sub-dataset corresponding to a specific task, and the task sets of the human and robot data are **non-overlapping**.

After training, we deploy P_{policy} on a real-world robot and evaluate its performance on **all tasks from D_{human}** to assess the effectiveness of motion transfer. This is defined as the **zero-shot** setting, since the evaluation tasks contain no corresponding robot data for training. We also evaluate the

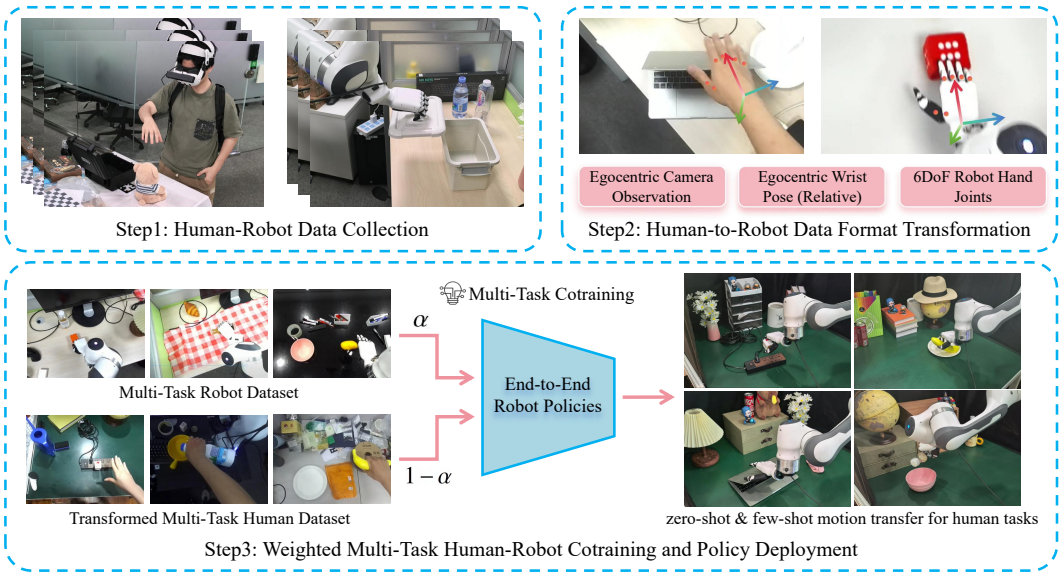


Figure 2: Illustration of our proposed *MotionTrans* framework, which consists of a human-robot data collection system, a pipeline for transforming human data into robot format, and a weighted human-robot multi-task cotraining strategy. After training, we enable the direct deployment of the trained policies to perform tasks in human datasets on real robots.

performance of **few-shot finetuning** setting, where a small number of robot demonstrations for the tasks from D_{human} are available to further finetune P_{policy}.

We define the input and output of our policies within the robot observation-action space $S = (I_t, P_t, A_t)$. This space S enable direct motion execution on real-world robots. At each timestamp t, the policy receives an egocentric RGB image $I_t \in \mathbb{R}^{H\times W\times 3}$ and proprioceptive states $P_t \in \mathbb{R}^{T_P\times D}$, where T_P is the history length and D is the state dimension. For simplicity, this work focuses on single-arm tasks (Figure 3), thus D corresponds to the concatenation of one robot wrist pose and one robot hand joint state (Figure 2(a)). The policy outputs an action chunk prediction $A_t \in \mathbb{R}^{T_A\times D}$ [2], where T_A denotes the action prediction horizon.

3.2 Human-Robot Data Collection System

For human-robot cotraining, we need to collect both robot and human data [13]. Next we describe the data collection system of these two type of data. The top-left side of Figure 2 illustrates the two types of data collection systems.

Human Data Collection with Portable VR Device. We extend ARCap [19] to build our human data collection system (Figure 2(a)), which incorporates a portable VR headset for recording hand keypoint positions and camera poses and an RGB camera for capturing the image stream I_t. The system enables the recording of hand keypoints position K_t and human wrist poses W_t in the I_t camera's coordinate frame. Collectors are instructed to minimize head motion to approximate the static camera setting of real robot hardware, although slight movements are tolerated [9].

Robot Data Collection with Teleoperation. Since our goal is to achieve direct human-to-robot motion transfer, the robot hardware need to match the functionality of the human arm and hand. To this end, we choose the combination of a single robot arm and a dexterous robot hand as our hardware platform (Figure 2(a)). The teleoperation system for this setup is based on Open-Television [1].

3.3 Human Data Transformation to Robot Format

As shown in the previous section, the raw human data collected from the VR device differs in format from robot data, which prevents it from being directly used for cotraining with deployable

robot policies [14, 15]. To address this, we propose directly transforming human data into **the robot's observation-action space** [1]. After transformation, the human data can serve as a form of "supplementary robot data" for training any mainstream end-to-end **robot** policy.

Transforming Observation-Action Space. The observation-action space of the robot includes three components: image observation I_t, proprioceptive state P_t, and action A_t (refer to Section 3.1). Both P_t and A_t are generated by stacking wrist poses W_t and hand joint states H_t. Next, we describe the design for these components: **(1) Image observation I_t:** We use **egocentric** view for both human and robot data, as shown in Figure 3. The use of the similar image view makes the spatial relationships of objects in the scenes similar for accomplishing similar tasks, thus enabling similar motions to achieve those tasks. **(2) Wrist poses W_t:** We use the **egocentric** camera coordinate system (camera captures I_t) for both human and robot data. This allows for the measurement of wrist poses in a unified coordinate system, ensuring that the spatial definitions of human and robot data are consistent. **(3) Hand joints state H_t:** we employ the dex-retargeting [52], an optimization-based inverse kinematics solver, to map human hand keypoints K_t to robot hand joint state H_t.

To further mitigate the difference between human data and robot data: (1) we slow down human data by a factor of 2.25 via poses and hand joints state interpolation; (2) we utilize action-chunk-based relative poses [2, 20] for wrist poses to reduce distribution mismatches between human and robot data. For instance, even if the robot's and human's hand positions differ in world space, their relative poses remain the same if they move forward at the same speed; (3) we encourage collectors to change viewpoints between trajectory recordings. This enhances the diversity of positional relationships between the camera view and the targeted manipulation objects, thereby encouraging policies to adapt to a larger distribution of hand poses and, consequently, a larger workspace for the robot.

The design above converts human data into the same format with robot, enabling us to direct replay human data on real-world robots. Based on the collection system, data transformation pipeline, and principles described above, we collect our *MotionTrans* human-robot datasets (Section 4.1 and Figure 3) for multi-task cotraining.

3.4 Weighted Multi-Task Human-Robot Cotraining

By unifying the observation and action spaces, we enable joint training of human and robot data under a shared end-to-end robot policy. This section introduce the multi-task policy architectures we use and how we train these policies.

End-to-End Multi-Task Policy Architectures. We explore two popular end-to-end policy architectures: (1) **Diffusion Policy (DP)** [2]: unlike the original single-task setup, we extend DP for multi-task training. Each task is associated with a learnable embedding, serving as a unique task condition. The visual encoder is replaced with DINOv2 [53] to enhance visual perception ability [17]. (2) **Vision-Language-Action model (π_0-VLA)**: we adopt network structure from [3], a policy architecture integrating large-scale pretrained Vision-Language Models [54] for multimodal perception and instruction following. We directly use instructions (listed in Appendix A) to assign tasks for π_0-VLA.

Weighted Human-Robot Cotraining. Our final step is to design a strategy to train multi-task policies with the processed human-robot dataset. Given the potential imbalance between human and robot data [9, 51], we adopt a weighted cotraining strategy similar to [55]. The training objective over the combined dataset $D = D_{\text{robot}} \cup D_{\text{human}}$ is defined as: $\mathcal{L}_D = \alpha \mathcal{L}_{D_{\text{robot}}} + (1-\alpha)\mathcal{L}_{D_{\text{human}}}$, where $\mathcal{L}$ denotes the loss computed on each dataset. In this paper, we set: $\alpha = \frac{|D_{\text{human}}|}{|D_{\text{human}}|+|D_{\text{robot}}|}$, where $|D_{\text{robot}}|$ and $|D_{\text{human}}|$ representing the dataset sizes. This weight ensures that the sum of the weights for human and robot data is equal, leading to the balance of these two data sources.

4 Experiments

In this section, we introduce our detailed experiment setting (Section 4.1) and results. The details of hardware platform, policies implementation and training could be found in Appendix B and Appendix C. We conduct experiments for both zero-shot (Section 4.2) and few-shot (Section 4.3)

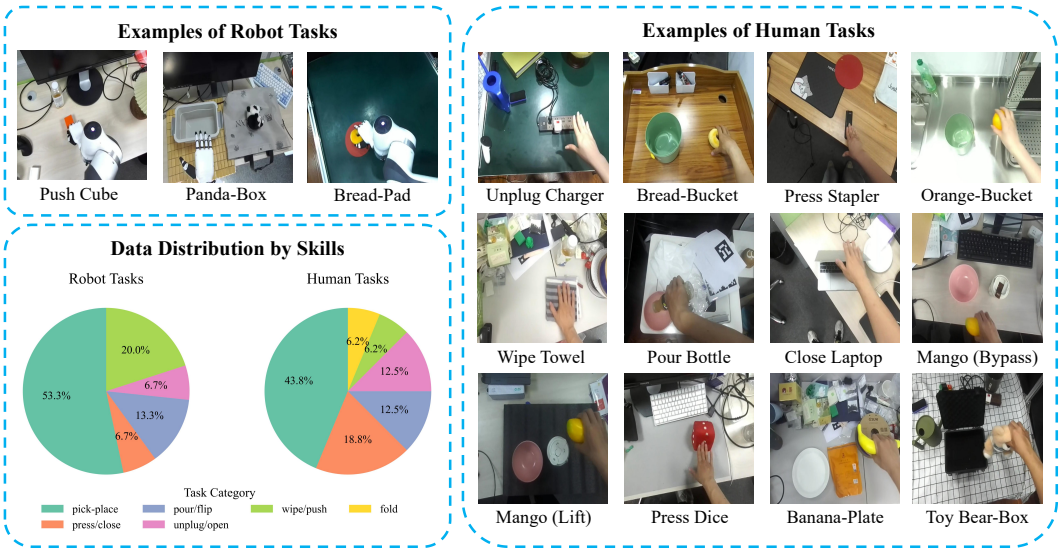


Figure 3: Illustration of the *MotionTrans Dataset*, which comprises 3,213 demonstrations spanning 15 human tasks and 15 robot tasks collected across more than 10 scenes. Detailed description and visualization of all 30 tasks are provided in Appendix A.

settings, as demonstrated in Section 3.1. Additionally results, including design ablation study, explanation of motion transfer mechanism and visual robustness verification could be found in appendix. Note that our conclusion focuses on **motion learning** and does not account for potential improvements in visual grounding or policy robustness from human data [14, 15].

4.1 Experiment Setup

***MotionTrans* Multi-Task Dataset.** Here we introduce the dataset used to train our policies, namely the *MotionTrans Dataset*, which includes 3,213 demonstrations across 15 human tasks and 15 robot tasks from more than 10 scenes. A brief summary of the dataset is shown in Figure 3. Each task contains between 40 and 150 demonstrations. We ensure the dataset covers a wide range of motions and skills, including pick-and-place, pouring, wiping, pushing, pressing, opening, etc. This variety has been proven crucial for successful motion transfer, as demonstrated in Appendix G. The complete task list and visualizations for all 30 tasks are provided in Appendix A.

To enhance the visual robustness of the policies [56] (Appendix H), such as robustness to different backgrounds and lighting conditions, we collect training data across 10+ scenes [17]. For robot tasks, about half of the data is collected in the *"green table scenes"* (the scenes for the examples of the "Bread-Pad" and "Unplug Charger" task in Figure 3). *This scene is also designated as the default scene for our evaluation.*

Evaluation Tasks and Metrics. Since our goal is to understanding the effectiveness of human-to-robot motion transfer, we focus on evaluating human tasks in this paper. Among all 15 tasks in human dataset, there are two tasks ("Fold Towel" and "Pour Milk Bottle") not been able to deploy to robot due to hardware design limitation. Therefore, we drop the results of these two tasks and focus on discussing other 13 tasks. The task list could be found in Figure 4 and Appendix A.

We use the *Success Rate (SR)* to evaluate the ability of policies to accomplish specific tasks. However, this metric alone is insufficient to reflect the effectiveness of motion transfer, as it ignores meaningful motion during task execution. To address this limitation, we define a *Motion Progress Score (Score)* to quantify the quality of policy motion for task completion. This metric enables a more precise evaluation of policy behavior. Detailed scoring rubrics for all tasks are provided in Appendix D. For each task, we conduct 10 rollouts and calculate the average results for both metrics. For clarity,

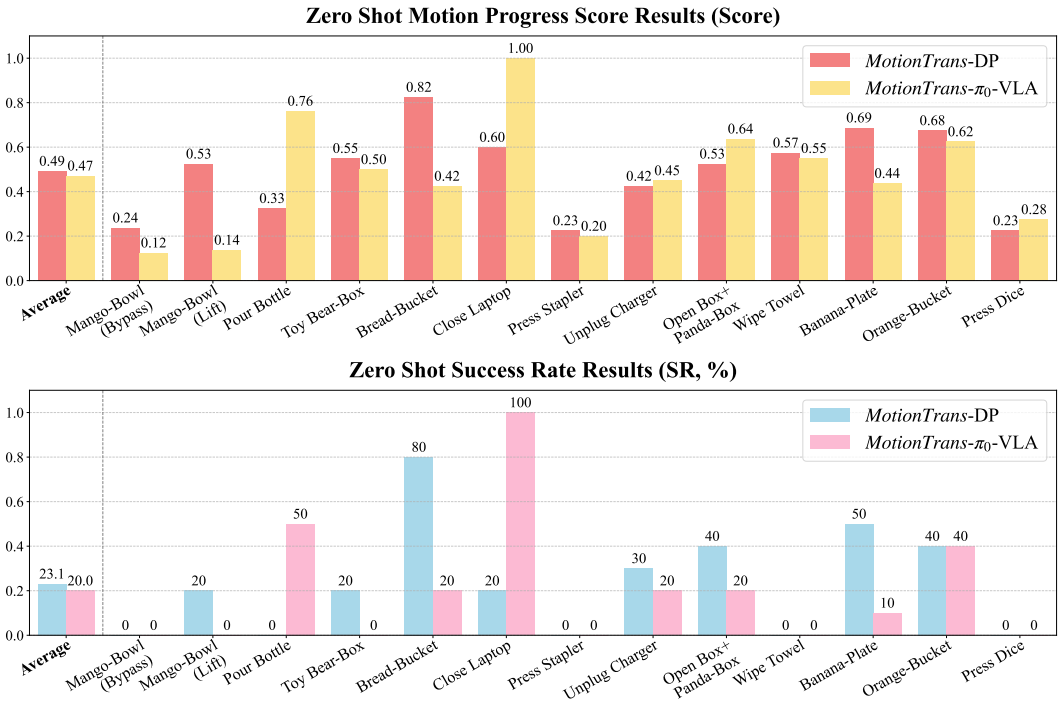


Figure 4: Results of *MotionTrans* in the zero-shot experiment setting. The results show that both Diffusion Policy (DP)[2] and π_0-VLA[3] achieve successful human-to-robot motion transfer. Even without any robot data for these human tasks, 9 tasks attain a non-zero success rate. For the remaining tasks, *MotionTrans* still generates meaningful motion for task accomplishment, as indicated by a non-trivial Motion Progress Score.

we normalize the Score to a [0,1] range for all presentations. We change the object arrangement for each rollout to cover a wide range of configurations of the task across the 10 rollouts.

4.2 Zero-shot Experiment

The goal of the zero-shot experiment is to verify the effectiveness of direct human-to-robot motion transfer. This means learning motions from human without collecting any robot data for these specific human tasks. We train the two selected end-to-end policies, Diffusion Policy (DP) and π_0-VLA (as mentioned in Section 3.4), using our *MotionTrans Dataset*. Subsequently, we directly deploy these policies to the real robot hardware and evaluate the performance of all **human tasks**. We seek to answer the following questions:

- (Q1.1) Can the robot directly learn to accomplish some tasks from human data, even without collecting any robot data for these tasks?
- (Q1.2) For tasks that cannot be accomplished, can the robot learn meaningful motion for task completion?
- (Q1.3) Is cotraining with robot data the key factor for achieving explicit motion transfer?
- (Q1.4) What is the difference in motion transfer effectiveness between different policy architectures?

In this paper, we focus on enabling human-to-robot transfer for mainstream end-to-end policies. Therefore, we do not compare against zero-shot intermediate representation-based methods such as Vid2Robot [57], General-Flow [12], EgoZero [45], ZeroMimic [33] etc., which are not compatible with such policies. Instead, our analysis centers on differences among end-to-end policy architectures (DP vs. π_0-VLA).

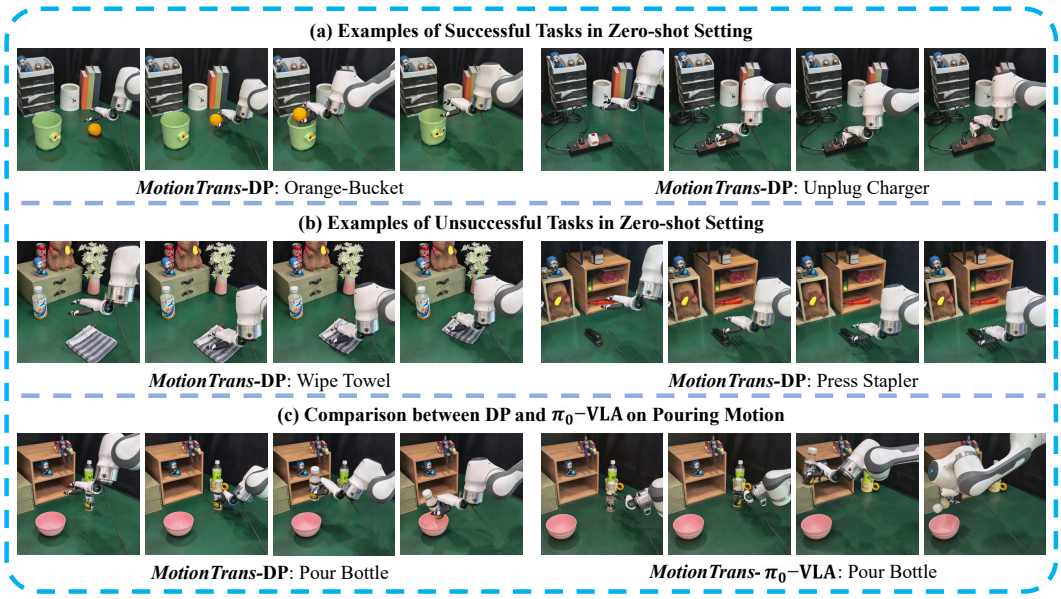


Figure 5: The visualizations for **zero-shot** human-to-robot motion transfer from our *MotionTrans* framework. All tasks shown here do not involve any robot data collection and are learned from human data. These results demonstrate that the *MotionTrans* enables explicit human-to-robot motion transfer for task completion through human-robot cotraining.

Results Overview. The results of the zero-shot experiment are shown in Figure 4. We observe that both DP and π_0-VLA learn meaningful motions for task completion, achieving an average Motion Progress Score of around 0.5. For some tasks, our method enables zero-shot task accomplishment. The average success rate on all tasks is approximately 20%. This prove that *MotionTrans* enable efficient human-to-robot motion transfer even for zero-shot setting.

(Q1.1) *MotionTrans* enables the policy to achieve non-trivial success rate across 9 tasks in the human dataset. As shown in Figure 4, there are 9 tasks that achieve a non-trivial success rate. The visualization of two examples could be found in the Figure 5(a) ("Orange-Bucket" and "Unplug Charger"). Among these tasks, pick-and-place tasks account for the vast majority. This can be attributed to the simplicity of pick-and-place motion and the large number of such tasks in our dataset. We want to mention that even if both the pick objects and place targets are not seen in robot tasks (e.g., the "Orange-Bucket" task, visualized on the left side of Figure 5(a)), this type of task-level transfer is still possible. Other accomplished tasks include motions such as pouring, unplugging, lifting, opening and closing (pressing), although some of these motions achieved only limited success rates under constrained object arrangement settings. In more general settings, the model often exhibits meaningful motion tendencies, though insufficient to accomplish the tasks.

(Q1.2) *MotionTrans* enables the robot to learn meaningful motion for task completion. We find that for tasks that cannot be directly accomplished in a zero-shot manner, *MotionTrans* still learns meaningful motions from human data (Figure 4). For instance, in the "Wipe Towel" task, both DP and π_0-VLA learn the motion of "push towel forward" to some extent (left side of Figure 5(b)). Moreover, we observe that human data provides a spatial location capability for almost all human tasks, which is represented as reaching the target manipulated objects (may only appearing in human data) to some extend. An example of this is the "Press Stapler" task in Figure 5(b). Although the stapler is not included in the robot data, the policy still demonstrates approaching behavior.

(Q1.3) Cotraining with robot data is the key factor for successful motion transfer. The "human-only" baseline in Figure 6 illustrates the results when only human data is utilized for Diffusion Policy (DP) training. It is evident that when robot data is not included for cotraining, the success rate across all tasks is 0% for zero-shot setting. The behavior score results can be found in Appendix E.

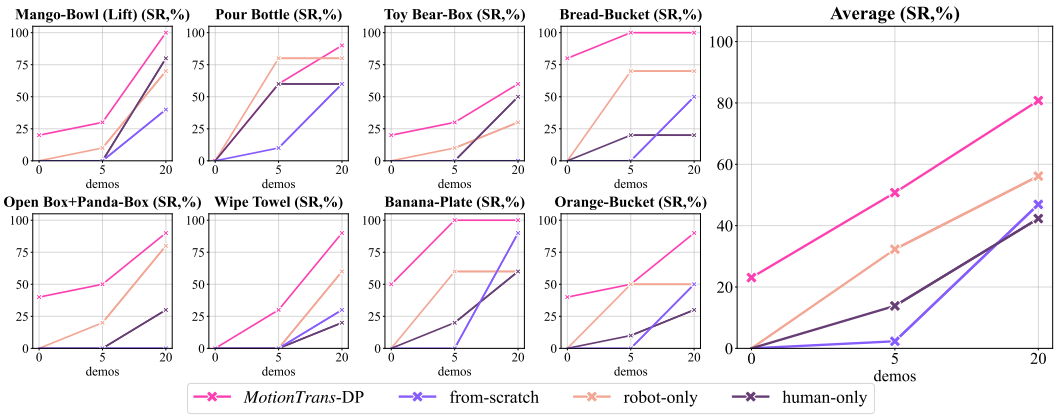


Figure 6: Results of the success rate for few-shot finetuning experiments. For readability, only the results of 8 example tasks are presented here. From these results, we conclude that both human and robot data during pretraining are important for improving finetuning performance.

Generally, the policy trained solely on human data exhibits random motion when deployed on the robot. This demonstrates that cotraining with robot data is essential for explicit human-to-robot motion transfer.

(Q1.4) DP and π_0-VLA each have their own advantages (manipulation precision and task adherence). As shown in Figure 4, no single model excels across all tasks. On average, the performance of the two models is nearly identical. However, we observe that different models demonstrate their strengths on different tasks. Generally, DP performs better than $\pi_0 - VLA$ in precise manipulation stage, such as grasping, and exhibits stronger spatial location capabilities. An example of evidence for this is that, for all pick-and-place tasks, the average grasping success rate of π_0-VLA is 20%, while DP achieves 67.5%. In contrast, π_0-VLA shows stronger instruction following for motion generation in more cases. For example, in the "Pour Bottle" task, we observed limited wrist rotation with DP, while π_0-VLA successfully performs the complete pouring action (Figure 5(c)). We hypothesize that this difference arises from a balance between visual perception and task semantic following. The model that focuses more on visual perception (DP) tends to achieve greater manipulation precision, whereas the model that emphasizes task semantics and instruction following (π_0-VLA) can adhere to task requirements more stringently.

4.3 Few-shot Experiment

In this section, we investigate whether motion transfer from human-robot cotraining can also enhance performance in a few-shot finetuning setting, where a limited number of robot demonstrations of human tasks are available for policy finetuning. Considering DP and π_0-VLA exhibit similar average performance in zero-shot experiments, we focus on DP architecture for computational resource efficiency in this part. We additionally collect 20 demonstrations for all human tasks in the default "green table" scenes, as mentioned in Section 4.1. Subsequently, we perform 5-shot and 20-shot **multi-task finetuning** [16] based on checkpoints previously trained on the *MotionTrans Dataset*. We aim to answer the following questions:

- (Q2.1) Will pretraining on the human-robot dataset, namely the *MotionTrans Dataset*, help improve policy finetuning performance?
- (Q2.2) What is the contribution of human data versus robot data for policy pretraining?
- (Q2.3) Does the conclusion vary based on the number of finetuning examples used?

To this end, we compared our method with three baselines to investigate the impact of different data components: (1) **"from-scratch"**, which means training policies without any pretraining; (2) **"robot-only"**, which entails pretraining solely on robot data from the *MotionTrans Dataset* before

finetuning; and (3) **"human-only"**, which consists of pretraining exclusively on human data prior to finetuning. Next, we present the detailed settings and results of few-shot experiments.

Result Overview. The results of the few-shot experiments are presented in Figure 6. For clarity, we only show the success rate results for a subset of tasks. The results for Motion Progress Score can be found in Appendix E. As observed, the average performance of the policies improves consistently with an increase in finetuning data. Next, we will conduct a detailed analysis to address the questions posed at the beginning of the experiment section.

(Q2.1) Pretraining on *MotionTrans Dataset* enable significant improvement for finetuning performance. As shown in Figure 6, policy pretrained on *MotionTrans Dataset* gains around 40% average success rate improvement compared to "from-scratch" baseline. These results prove that cotraining on human-robot data could provide useful motion prior [58] for downstream finetuning.

(Q2.2) Both human and robot data during pretraining are crucial for enhancing performance. From Figure 6, we can see that policy pretrained on both human and robot data (*MotionTrans*) shows a significant advantage compared to human-only or robot-only pretraining. Besides, robot-only pretraining outperforms human-only pretraining on average. This is expected, as our robot pretraining data share the same embodiment as the deployed robot, minimizing the embodiment gap between training and inference [51]. All these results indicate that robot data play a non-negligible role during policy pretraining.

(Q2.3) Human-robot pretraining is more effective as the number of finetuning data decreases. Finally, we analyze the impact of pretraining with varying amounts of finetuning data. As shown in Figure 6, the improvements are much larger in the 5-shot setting compared to the 20-shot setting. Moreover, when 20 finetuned demonstrations are available, the advantage of robot-only pretraining becomes minimal, and the benefit of human-only pretraining disappears. However, in the 5-shot setting, all pretraining methods show a significant advantage over the from-scratch baseline.

4.4 Other Results

We leave the ablation study of the key designs in Appendix F. We also conduct a case study in Appendix G to understand the mechanism and reasons of motion transfer. (1) Transfer occurs through using human motion to encourage motion interpolation of robot data to formulate new motions for human task (motion generation). (2) The visual encoder learns to attend to the target object in the human data (visual perception). (3) The transfer performance improves with a wider range of motion and task coverage (scaling trend). Finally, we verify the visual robustness of our experiment results against background in Appendix H.

5 Conclusion

In this paper, we propose *MotionTrans*, a framework that achieves motion-level learning from human data for end-to-end robot policies. The experiments show that our method achieves explicit human-to-robot motion transfer in a zero-shot setting and significantly improves finetuning performance in a few-shot setting. We also reveal the key factor for explicit motion transfer: cotraining with robot data. We hope that the new motion-centric insights that we propose could enhance the utilization of human data in robot policy learning in more effective ways.

Limitations and Future Directions. Our largest limitation is that the height perception ability of the policies is still limited, which causes them to sometimes fail to reach the correct height when considering in-the-wild scenes. This limitation arises from our monocular egocentric perception setting, which may be addressed by adding wrist camera for both human and robot hardware platforms [51, 59]. Another limitation is that our study is still limited to self-collected human dataset. Extending motion-level learning to larger, internet-scale datasets, as in [15], is left for future work.

References

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

[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[4] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[5] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[6] Y. Ji, H. Tan, J. Shi, X. Hao, Y. Zhang, H. Zhang, P. Wang, M. Zhao, Y. Mu, P. An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1724–1734, 2025.

[7] E. Zhou, J. An, C. Chi, Y. Han, S. Rong, C. Zhang, P. Wang, Z. Wang, T. Huang, L. Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.

[8] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. pi_0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[9] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[10] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[11] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[12] C. Yuan, C. Wen, T. Zhang, and Y. Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024.

[13] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv:2410.24221*, 2024.

[14] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, H. Yin, S. Liu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[15] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[16] H. Bi, L. Wu, T. Lin, H. Tan, Z. Su, H. Su, and J. Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. *arXiv preprint arXiv:2507.23523*, 2025.

[17] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[18] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint arXiv:2507.05331*, 2025.

[19] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024.

[20] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

[21] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[22] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[23] F. Lin, R. Nai, Y. Hu, J. You, J. Zhao, and Y. Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[24] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022.

[25] C. Yuan, G. Chen, L. Yi, and Y. Gao. Self-supervised monocular 4d scene reconstruction for egocentric videos. *arXiv preprint arXiv:2411.09145*, 2024.

[26] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

[27] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[28] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[29] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

[30] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[31] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[32] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.

[33] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. *arXiv preprint arXiv:2503.23877*, 2025.

[34] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024.

[35] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

[36] S. Patel, S. Mohan, H. Mai, U. Jain, S. Lazebnik, and Y. Li. Robotic manipulation by imitating generated videos without physical demonstrations. *arXiv preprint arXiv:2507.00990*, 2025.

[37] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[38] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

[39] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025.

[40] H. Kim, J. Kang, H. Kang, M. Cho, S. J. Kim, and Y. Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. *arXiv preprint arXiv:2505.08787*, 2025.

[41] H. Zhou, R. Wang, Y. Tai, Y. Deng, G. Liu, and K. Jia. You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations. *arXiv preprint arXiv:2501.14208*, 2025.

[42] X. Zhu, Y. Liu, H. Li, and J. Chen. Learning generalizable robot policy with human demonstration video as a prompt. *arXiv preprint arXiv:2505.20795*, 2025.

[43] C. Tang, A. Xiao, Y. Deng, T. Hu, W. Dong, H. Zhang, D. Hsu, and H. Zhang. Functo: Function-centric one-shot imitation learning for tool manipulation. *arXiv preprint arXiv:2502.11744*, 2025.

[44] S. Park, H. Bharadhwaj, and S. Tulsiani. Demodiffusion: One-shot human imitation using pre-trained diffusion policy. *arXiv preprint arXiv:2506.20668*, 2025.

[45] V. Liu, A. Adeniji, H. Zhan, S. Haldar, R. Bhirangi, P. Abbeel, and L. Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025.

[46] S. Wang, J. You, Y. Hu, J. Li, and Y. Gao. Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation. *arXiv preprint arXiv:2501.14400*, 2025.

[47] J. Engel, K. Somasundaram, M. Goesele, A. Sun, A. Gamino, A. Turner, A. Talattof, A. Yuan, B. Souti, B. Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

[48] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[49] Y. Niu, Y. Zhang, M. Yu, C. Lin, C. Li, Y. Wang, Y. Yang, W. Yu, T. Zhang, Z. Li, et al. Human2locoman: Learning versatile quadrupedal manipulation with human pretraining. *arXiv preprint arXiv:2506.16475*, 2025.

[50] M. Lepert, J. Fang, and J. Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025.

[51] T. Tao, M. K. Srirama, J. J. Liu, K. Shaw, and D. Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

[52] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023.

[53] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[54] A. Steiner, A. S. Pinto, M. Tschannen, D. Keysers, X. Wang, Y. Bitton, A. Gritsenko, M. Minderer, A. Sherbondy, S. Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[55] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels. *arXiv preprint arXiv:2503.22634*, 2025.

[56] C. Yuan, S. Joshi, S. Zhu, H. Su, H. Zhao, and Y. Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. *arXiv preprint arXiv:2503.18738*, 2025.

[57] V. Jain, M. Attarian, N. J. Joshi, A. Wahid, D. Driess, Q. Vuong, P. R. Sanketi, P. Sermanet, S. Welker, C. Chan, et al. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. *arXiv preprint arXiv:2403.12943*, 2024.

[58] W. Ye, Y. Zhang, M. Wang, S. Wang, X. Gu, P. Abbeel, and Y. Gao. Foundation reinforcement learning: towards embodied generalist agents with foundation prior assistance. 2023.

[59] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025.

[60] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022.

[61] G. H. An, S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang. Charuco board-based omnidirectional camera calibration method. *Electronics*, 7(12):421, 2018.

[62] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[63] G. Li, Y. Lyu, Z. Liu, C. Hou, J. Zhang, and S. Zhang. H2r: A human-to-robot data augmentation for robot pre-training from videos. *arXiv preprint arXiv:2505.11920*, 2025.

[64] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[65] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Appendix

A Details of *MotionTrans Dataset* and All Tasks

Here we describe the details of all tasks in *MotionTrans Dataset*. For simplicity, we name pick-place task with "pick object-place target" format, and name other task with "verb noun" format in the main paper. For tasks with multiple steps, we name it as "step1+step2" format.

The visualization and descriptions / VLA-prompt of all 15 human tasks could be found in Figure 7 and Table 1. All 15 robot tasks could be found in could be found in Figure 8 and Table 2.

As mentioned in the main paper, to enhance the visual robustness of the policies [56] (Appendix H), such as robustness to different backgrounds and lighting conditions, we collect training data across various scenes [17]. Each human task is collected in at least 4 different scenes. For robot tasks, about half of the data is collected in the default *"green table scenes"*, with random disturbance objects placed on the table for approximately 80% of the data. The other half of the robot tasks is collected in at least 4 different scenes.

B Hardware Platform

For the robot hardware (Figure 10a(c)), we use a Franka Emika robot arm [60] in combination with a 6DoF Inspired Dexterous (Right) Hand [1]. This combination mimics the functionality of a human right hand and arm. The robot is mounted on a movable lift table to facilitate data collection in various locations. A ZED2 camera is fixed to the table in an egocentric view to provide an image observation stream. The recorded images are cropped to 640x480 resolution. The VR device used for teleoperation is the Meta Quest 3 [1]. Calibration between the robot base and the robot perception camera is achieved through the DROID platform codebase [4].

For human data collection (Figure 10a(a)), we use the Meta Quest 3 as our VR headset. To ensure consistency in image observations, we also employ a ZED2 camera to record RGB images and perform image cropping. We designed a 3D-printed element to link the VR headset and the camera [19]. This device is connected to a portable workstation, such as a laptop, for data storage. Calibration between the camera and the VR device is accomplished through a combination of ArUco code calibration [61] and VR anchor location (based on VR APP). To ensure data quality, we provide real-time feedback on the camera's view frustum and hand positioning to guide the collector during data acquisition (Figure 10a(b)).

C Policies Implementation

For the robot observation-action space (Section 3.1), we set the proprioception history $T_p = 2$ and the action horizon $T_A = 16$. The representation of the rotation component of wrist poses is chosen as the first two rows of the rotation matrix, as demonstrated in [2]. For policy control, we use 10 fps for both data collection and policy inference. For Diffusion Policy (DP) backbone, the task-embedding dimension is set as 16. The proprioception state is encoded via a 4-layer MLP. The DINOv2 vision encoder utilizes DINOv2-base pretrained checkpoints [53], and during training, we unfreeze the weights of the DINOv2-base encoder. We first concatenate the task embedding with the features from the vision and proprioception encoder, and then input the concatenated features into the U-Net-based Diffusion head for action generation [2]. For π_0-VLA, we load π_0-droid pretrained checkpoints [62] before training. Since the target hardware of this checkpoint is incompatible with our robot hardware, direct deployment results in a 0% success rate. The model's performance is derived from training within our *MotionTrans* framework.

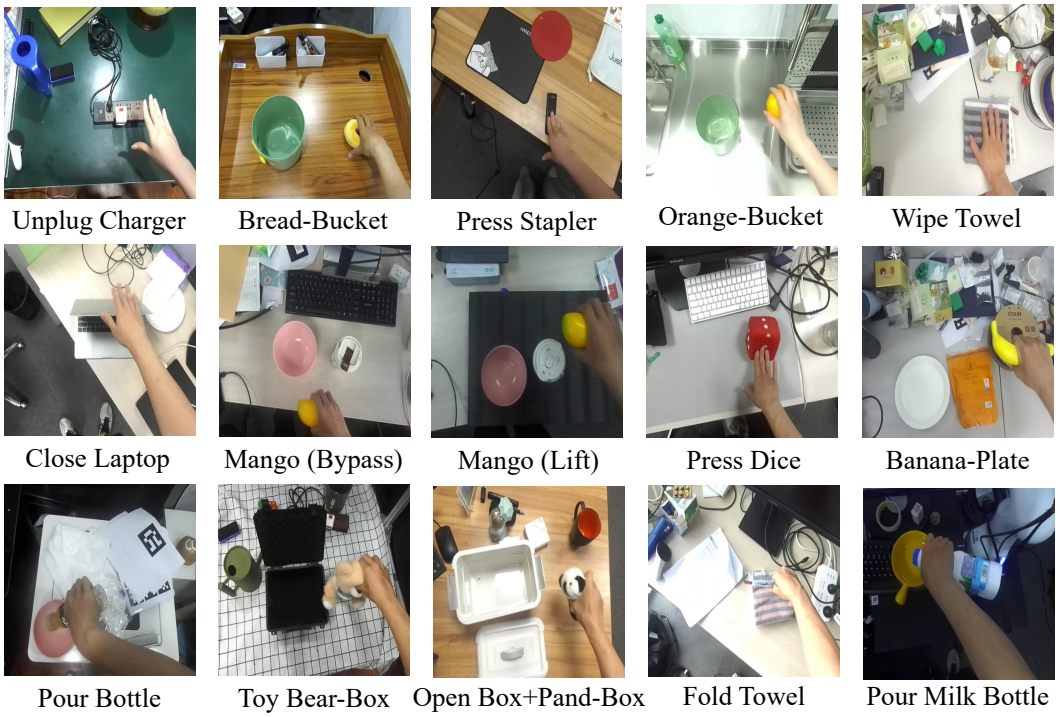


Figure 7: The visualizations of all 15 human tasks in the egocentric view.

Human Tasks	**Description / VLA-prompt**
Unplug Charger	unplug the white charger.
Bread-Bucket	drop bread to the green bucket.
Press Stapler	press the stapler.
Orange-Bucket	put orange to the green bucket.
Wipe Towel	wipe blue towel on the table and push it to the bulky bottle.
Close Laptop	close silver laptop.
Mango-Bowl (Bypass)	put mango to pink bowl while avoiding obstacle by bypassing.
Mango-Bowl (Lift)	put mango to the pink bowl while avoiding obstacle by lifting.
Press Dice	press red dice to make it rotation.
Banana-Plate	put banana to the white plate.
Pour Bottle	pour bottle to the pink bowl.
Toy Bear-Box	put toy bear to the black box.
Open Box + Pand-Box	first open the white cap style box then put toy panda to the box.
Fold Towel	fold the blue towel.
Pour Milk Bottle	pour milk bottle to the yellow pan.

Table 1: All 15 human tasks with descriptions (VLA-prompt).

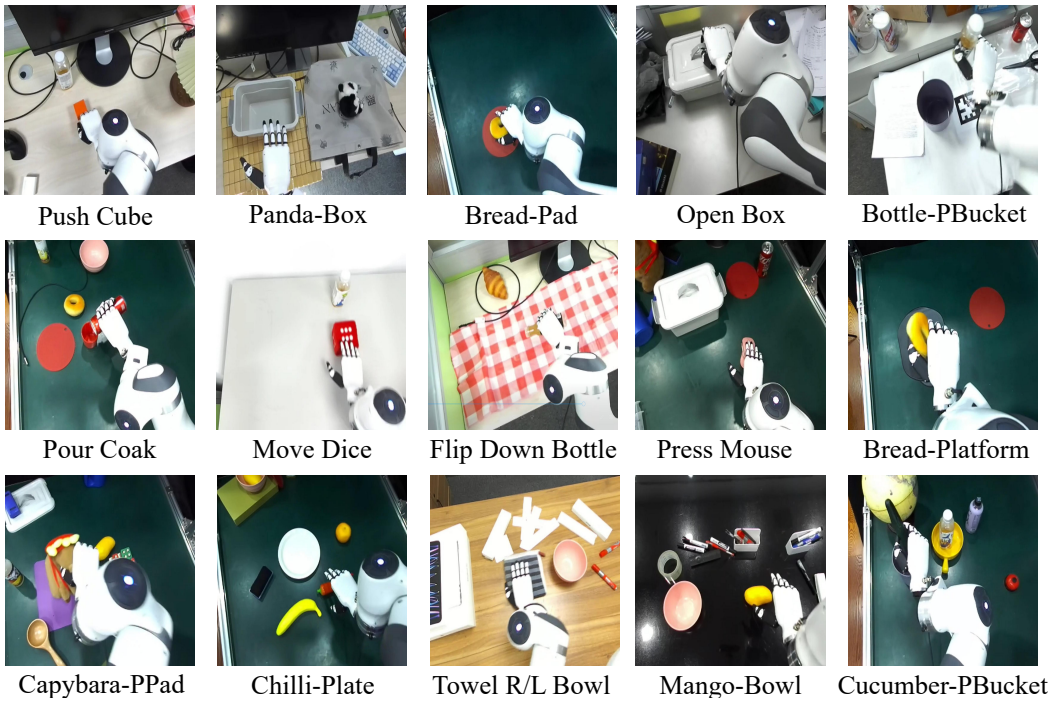


Figure 8: The visualizations of all 15 robot tasks in the egocentric view.

Robot Tasks	**Description / VLA-prompt**
Push Cube	push orange cube to the bulky bottle.
Panda-Box	put toy panda to the box.
Bread-Pad	put bread to the red pad.
Open Box	open the white cap style box.
Bottle-PBucket	drop black bottle to purple bucket.
Pour Cola	pour cola to the red cup.
Move Dice	move red dice to the bulky bottle.
Flip Down Bottle	flip down the black bottle.
Press Mouse	press the pink mouse.
Bread-Platform	put bread to the high black platform.
Capybara-PPad	put Capybara to the purple pad.
Chilli-Plate	put chilli to the white plate.
Towel R/L Bowl	wipe blue towel on the table and push it left or right to the pink bowl.
Mango-Bowl	put mango to the pink bowl.
Cucumber-PBucket	put cucumber to purple bucket.

Table 2: All 15 robot tasks with descriptions (VLA-prompt).

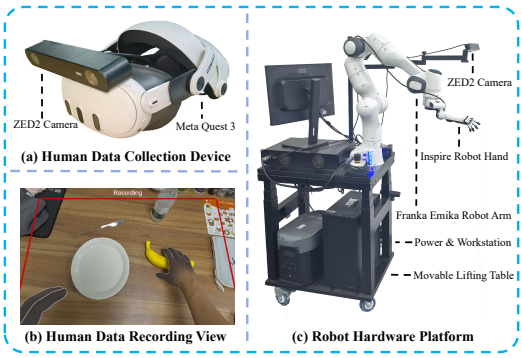
Figure 9: Results of the Motion Progress Score for few-shot finetuning experiments. For readability, only the results of 8 example tasks are presented here.

D Rubrics of Motion Progress Score

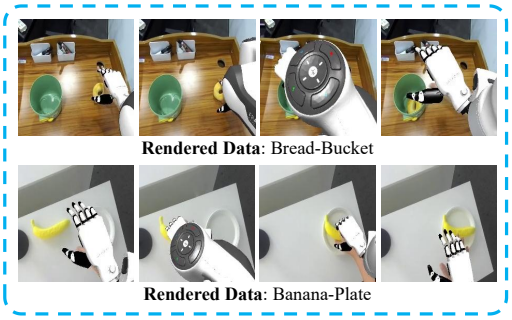
Table 3 provides the detailed rubrics for our Motion Progress Score metric. The scores are allocated to the different motions / stages required to complete the task, with a maximum score of 8 points.

Human Tasks	**Rubrics of Motion Progress Score**
Mango-Bowl (Bypass)	(1) show reach-grasp; (1) success grasp; (2) show bypassing; (2) success bypassing; (1) show reach-put; (1) success put;
Mango-Bowl (Lifting)	(1) show reach-grasp; (1) success grasp; (1) show lifting; (2) success lifting; (2) show down-putting; (1) success put;
Pour Bottle	(1) show reach-grasp; (1) success grasp; (2) show rotation; (2) success pouring; (2) good pour position;
Toy Bear-Box	(2) show reach-grasp; (2) success grasp; (2) show reach-put; (1) success put; (1) good put;
Bread-Bucket	(1) show reach-grasp; (1) success grasp; (2) show reach-put; (2) success put; (2) good put height;
Close Laptop	(2) show reach-press; (2) press finish < 30 degrees; (2) press finish < 15 degrees; (2) press finish = 0 degrees;
Press Stapler	(2) show reach-press; (2) success contact; (2) good contact; (2) press down;
Unplug Charger	(2) show reach-grasp; (1) success grasp; (1) show lifting; (2) success unplug; (2) still holding after unplug;
Open Box + Panda-Box	(2) open the white box; (1) continue; (1) no stop; (1) reach the panda; (1) success grasp the panda; (2) success put;
Wipe Towel	(2) show down-press; (2) success press; (2) show pushing (including retry); (2) success pushing;
Banana-Plate	(1) show reach-grasp; (2) success grasp; (2) show reach-put; (2) success put; (1) good put height;
Orange-Bucket	(1) show reach-grasp; (2) success grasp; (2) show reach-put; (2) success put; (1) good put height;
Press Dice	(1) show reach-press; (1) success contact; (2) show down-press; (2) press > 5 cm; (2) success press to make it rotate;

Table 3: The rubrics of Motion Progress Score for all 13 evaluated human tasks.



(a) Figure 10a: Illustration of our hardware system, which includes a human VR-based data collection device and a single-arm robot platform. A screenshot of the VR device during human data collection is also provided.



(b) Figure10b: The visualizations of the rendered RGB observations for the **w/ Visual Rendering** variant in design ablation (Appendix F).

E Results of Motion Progress Score for Few-shot Experiment

The results of Motion Progress Score for few-shot experiment are shown in Figure 9. The conclusion drawn from the Motion Progress Score aligns with that from the Success Rate (Section 4.3).

F Design Ablation

We conduct an ablation study on the key designs of *MotionTrans*.We find that when considering *motion-level* learning, the effectiveness of some designs may differ compared to *visual-level* human data learning [9, 13, 14, 15, 50, 63]. We compare three variants of *MotionTrans* in zero-shot setting experiments. The backbone chosen for all models is Diffusion Policy (DP):

- **w/ Abs Pose**: We replace the action-chunk-based relative pose [64] with absolute egocentric pose for wrist label.
- **w/ Action Norm** [13, 51]: We use independent action and proprioception normalization for human and robot data before policy training.
- **w/ Visual Rendering** [48, 50, 63]: We first replay robot data in simulation, then crop the rendered robot and paste it to the original RGB image observation. For simplicity, we ignore the segmentation of the human hand and the inpainting of the original human hand region, as demonstrated in [48]. Visualizations of the rendered results are shown in Figure 10b.

Results are shown in Table 4. We observe that **w/ Abs Pose** and **w/ Action Norm** dramatically decrease the performance of human-to-robot motion transfer. For **w/ Abs Pose**, this increases the distribution difference between human and robot actions, prohibiting motion transfer, as discussed in Section 3.3. For **w/ Action Norm**, performance drops because it creates a discrepancy in normalization between policy training and deployment. This contrasts with the phenomenon observed in visual robustness evaluations as demonstrated by previous works [13, 51]. When directly learning new motions and skills from human data, it is preferable to keep action normalization consistent between training and inference.

For **w/ Visual Rendering**, we find that performance is nearly the same as the non-rendered version. We believe this is because, although the rendered results may appear somewhat realistic to humans, they still contain many hints and shortcuts for neural networks, which do not differ much from original human videos. One potential solution is to also conduct inpainting during policy inference [48], but may lead to additional computational overhead and policy delay.

	Score	SR (%)
w/ Abs Pose	0.370	10.0
w/ Action Norm	0.341	8.4
w/ Visual Rendering	0.475	**23.1**
MotionTrans-DP	**0.492**	**23.1**

Table 4: Ablation results of design choices for *MotionTrans*. The results are averages across all evaluated human tasks. Analysis are demonstrated in Appendix F.

	Score	SR (%)
H-bucket	0.0	0
H-bucket + R-pad	0.275	0
H-bucket + R-platform	0.5	30
H-bucket + R-pad + R-platform	0.625	40
H-bucket + R-pad + R-platform + PP-set	0.75	70
all data (*MotionTrans*)	0.825	80

Table 5: The results of the case study for the "Bread-Bucket" task in zero-shot setting, including outcomes from training on different subsets of *MotionTrans Datasets*. Detailed analysis could be found in Section G.

G How Transfer Happening: A Case Study

In this section, we design a case study to delve deeper into the mechanism of human-to-robot motion transfer. We choose the task "Bread-Bucket" for our case study, as it already demonstrates a high success rate (80%) in zero-shot settings, indicating effective motion transfer for this task. We conduct the case study by **down-sampling the number of tasks**: we train policies on different subsets of *MotionTrans* and compare their performance, thereby gaining insights into how different training tasks affect the results. The subsets we use are:

- **(Human) Bread-Bucket:** the task we evaluate for the case study, referred to as "H-bucket" in later experiments.
- **(Robot) Bread-Pad:** placing bread on a very thin red pad, referred to as "R-pad."
- **(Robot) Bread-Platform:** placing bread on a very high black platform, referred to as "R-platform."
- **(Human-Robot) Pick-Place subset:** including four tasks: robot "Mango-Bowl," robot "Capybara-Purple Pad," human "Banana-Plate," and human "Toy Bear-Box." This subset is referred to as "PP-set" later.

Visualization and the height (for later analysis) of the key objects used in this experiment are shown in Figure 12a. We then train Diffusion Policy (DP) on different subsets, and the evaluation results for the "Bread-Bucket" task are shown in Table 5. By analyzing the results, we can conclude that:

(Conclusion 1: Action) Transfer occurs through motion interpolation of robot data to formulate new motions for human tasks. We first analyze the results of the first four rows of Table 5. The trajectory visualizations can be found in Figure 11. We observe that when only training on human data, the policy becomes confused with the newly deployed robot embodiment. When cotraining with one robot task (R-pad or R-platform), the robot tends to adapt to the placement height in the robot data rather than the height of the bucket. Only after cotraining with both R-pad and R-platform, which allows for interpolation of bucket height placement, does the policy begin to exhibit bucket-height-aware motion. This shows that rather than directly expecting to learn extrapolation from human data, we should cotrain with robot data and use motion in human data to facilitate robot motion interpolation for generating motions for these human tasks.

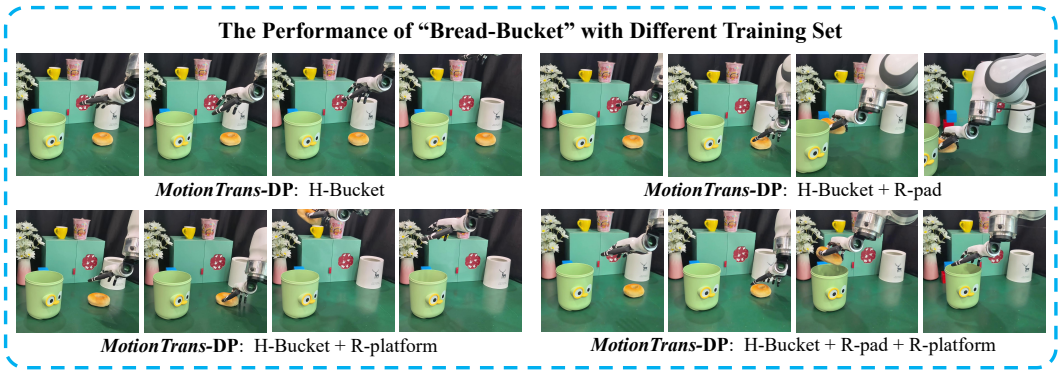
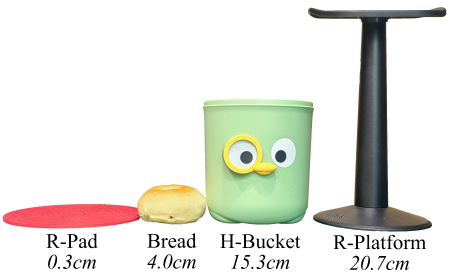
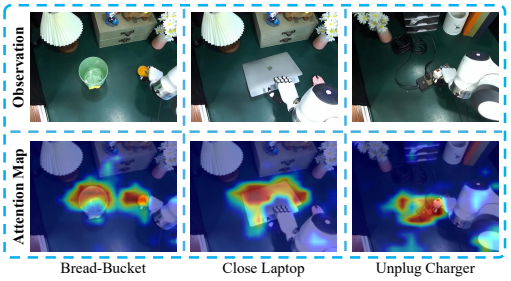


Figure 11: The visualizations of the *MotionTrans*-DP results for the "Bread-Bucket" task, trained on various combinations of human and robot tasks. By analyzing these results (Section G), we suggest that motion transfer occurs through the use of motion in human data to support robot motion interpolation for generating motions for these human tasks.



(a) Figure 12a: The visualizations of key objects used in the "Bread-Bucket" case study are presented here. The height of each object is labeled beneath it.



(b) Figure12b: The visualization of the attention map from the DINO encoder [53] for *MotionTrans*-DP, based on the Grad-Cam toolkit [65]. This shows that the vision encoder learns to focus on the target manipulation objects for tasks in human datasets, even when the embodiment changes to a robot during inference.

(Conclusion 2: Vision) The visual encoder learns to attend to the target object in the human data. Next, we analyze motion transfer from visual perception perspective. We visualize the attention map of the DINOv2 encoder [53] for "Bread-Bucket" and other tasks in Figure 12b via Grad-Cam [65]. We can see that, through training on human data, although the embodiment in image observation changes during policy deployment, the model still learns to attend to target objects to some extent. This explains why the policy gains the ability to locate target objects during robot deployment, even if these objects have only been seen in human data.

(Conclusion 3: Scaling) The transfer performance improves with a wider range of motion and task coverage. Building on Conclusions 1 and 2, we hypothesize that with a wider range of motion and task coverage, the policy may develop a stronger ability for motion interpolation and visual attention, thus leads to a better transfer performance. We verify this through the last three rows of Table 5. As the results show, with more task coverage, the performance improves steadily. This indicates that human-to-robot motion transfer exhibits a scaling trend to some degree.

H Visual Background Robustness

Finally, we verify the visual robustness of our experiment results against scene background [14]. We change the background from our default "green table" scenes (mentioned in Section 4.1) to a new scene, as shown in Figure 13, and evaluate Diffusion Policy (DP) performance for both zero-shot and 20-shot settings. The results are shown on the right side of Figure 13. We observe that although the performance drops slightly, it still maintains a non-trivial Motion Progress Score and success rate.

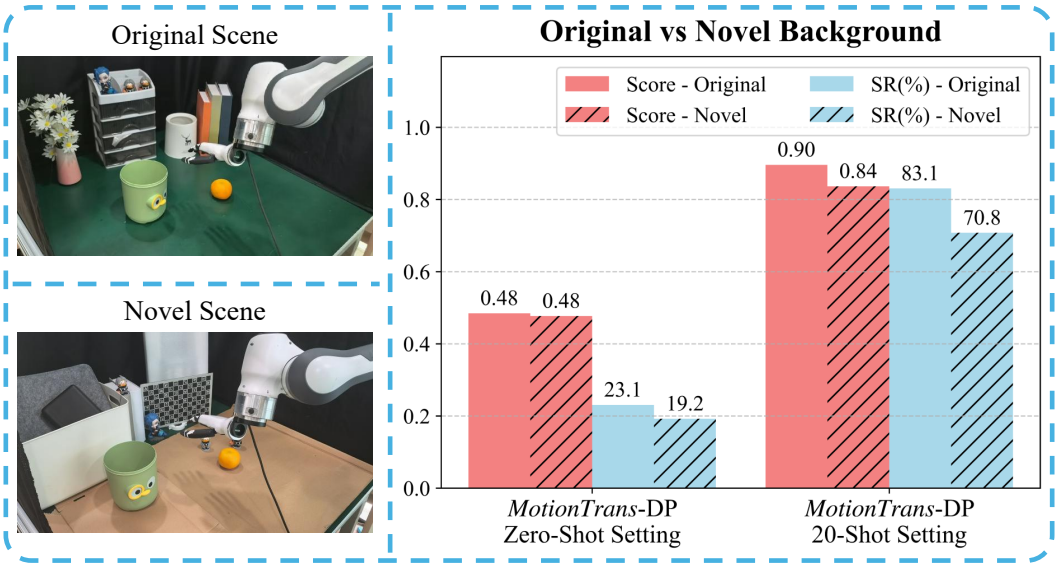


Figure 13: Illustration of the visual background robustness experiment and results. For the novel background, the performance drops slightly but remains at a persuasive level. This prove the robustness of our motion transfer results.

This proves the robustness of our results on motion-level human data learning. Note that this does not mean we achieve in-the-wild manipulation ability [51], which is not the main focus of this paper and will be discussed in the limitations section.