Learning to Transfer Human Hand Skills for Robot Manipulations

Sungjae Park*,1,2, Seungho Lee*,2, Mingi Choi*,2, Jiye Lee[2], Jeonghwan Kim[2], Jisoo Kim[2], Hanbyul Joo[2]

***Abstract*— We present a method for teaching dexterous manipulation tasks to robots from human hand motion demonstrations. Unlike existing approaches that solely rely on kinematics information without taking into account the plausibility of robot and object interaction, our method directly infers plausible robot manipulation actions from human motion demonstrations. To address the embodiment gap between the human hand and the robot system, our approach learns a joint motion manifold that maps human hand movements, robot hand actions, and object movements in 3D, enabling us to infer one motion component from others. Our key idea is the generation of pseudo-supervision triplets, which pair human, object, and robot motion trajectories synthetically. Through real-world experiments with robot hand manipulation, we demonstrate that our data-driven retargeting method significantly outperforms conventional retargeting techniques, effectively bridging the embodiment gap between human and robotic hands.**
https://rureadyo.github.io/MocapRobot/

I. Introduction

Recent advances in imitation learning (IL) via expert demonstrations have significantly improved dexterous manipulation with multi-fingered robotic hands [1]–[5]. These demonstrations typically come from robotic teleoperation, where a human teleoperates a robot hand using motion capture gloves or a vision-based hand estimation module. While such data provide physical plausibility of robot actions, collecting such demonstrations via teleoperation is often costly, time-consuming, and requires sophisticated skills to operate the robot hardware, which can vary in performance across operators due to the structural limitations of the system. In contrast, demonstrations via human motion capture offer a more natural and convenient alternative. Recent advancements in vision-based methods promise more accessible solutions [6]–[8], allowing users to perform tasks casually and potentially generating a larger volume of data. Motion capture data also provides rich hand information about how to manipulate the object, which is crucial for dexterous manipulation.

However, transferring human motion demonstrations to robots is not straightforward due to the embodiment gap between human and robotic hands. Differences in skeletal structure, hand size, and the forces that can be applied present significant challenges in directly applying imitation learning strategies. Consequently, traditional retargeting methods that transfer human demonstrations to robotic hands via, for example, kinematics-based [1], [5], [9] alignment often yield suboptimal results, leading to task failures. An ideal retargeting should be **generalizable** to different hand/object motions to benefit from the scalability of human motion demonstrations, and also output **physically plausible actions** for the robot hand under such embodiment gap.

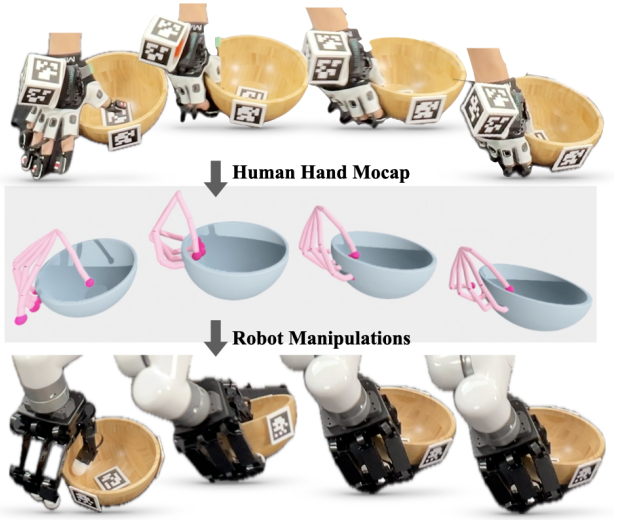

Fig. 1: Our model learns a **human-to-robot retargeting model** using an **unpaired** (i.e., object may move differently) human mocap and robot teleoperation dataset.

In this paper, we propose a novel approach to inferring plausible robot hand actions from human motion demonstrations, through a learning-based retargeting method. Specifically, our aim is to find the mapping between the robot hand actions and human hand motion to achieve the same target object motion. To achieve this, we formulate the problem within a supervised learning framework, learning a joint manifold space among human hand motion, robot action, and 3D object movements in a data-driven manner. The primary challenge in learning such mapping is the lack of available paired datasets for common desired actions. To address this, we introduce a method to synthetically generate paired human-robot grasping data by combining separately captured human motion capture demonstrations and teleoperation data on the same target object. To this end, our method can infer the effective robot action trajectory from human manipulation demonstration by finding an optimal latent code of the manifold space toward the provided human and object motion trajectories. Through extensive evaluations, we demonstrate the effectiveness of our approach in solving complex dexterous manipulation tasks.

II. Related Work

Robotic Teleoperation. In recent years, robotic teleoperation has emerged as a common data source for robotics. Several work proposed a teleoperation system across different robotic platforms, ranging from dexterous robot hand [1]–[4], mobile robot [10], bimanual robot [11]–[14], and humanoid

*equal contribution
[1], Carnegie Mellon University, [2] Seoul National University

robot [15]–[17]. Such systems often consist of a perception module which estimates human teleoperator's motion and a retargeting algorithm that kinematically maps human motion to robot actions. While teleoperation data contains rich information for training robotic policy via imitation learning, it has two major disadvantages: a necessity of hardware system (i.e., robot and teleoperation device such as VR devices [11], [18],exoskeleoton [12] and additional linkage system [19]) and an embodiment gap between human teleoperator and the robot. Among two, the latter often receives less attention while significant. Specifically, as the human teleoperateor indirectly interacts with the environment through the system, the teleoperator should fill the embodiment gap via visual feedback(i.e., see whether the robot is interacting with the environment in a desired way) during data collection. If the robot is not acting as intended, the teleoperator should implicitly adapt teleoperation strategy over time, making teleoperation less scalable when it comes to complex manipulation tasks.

Retargeting Human Hand to Robot Hand. Retargeting human's motion or interaction with the environment to that of robot has been a challenging research topic [20]–[24]. The most relevant to ours is retargeting human hand motion to robot hand. [1], [3], [5], [9] use an optimization algorithm via kinematic loss, which aim to match certain human joints' position(e.g. fingertips) with that of corresponding robot joints. Temporal consistency loss and self-collision loss are also taken into account optionally. [2] learn an energy model instead of performing optimization with a similar learning objective. While such methods work well for teleoperation, it may fail when the aim is to reproduce the interaction between human hand and the object, as matching joint positions doesn't necessarily result in same interactions. To consider the interaction between human hand and the object, [25] additionally considers contact heatmap given human hand and object mesh, such that the retargeted robot hand can grasp the same object similar to human, while being limited to grasping. [26], [27] also consider contact regions between human hand and object and try to match it within target robot and same object, but requires human expert labels, such as the center of contact region within the target robot, corresponding points between human hand and robot, etc. Although matching contact regions may result in a more physically plausible robot motion along the object, it does not guarantee the contact region from human mocap to be perfectly matched, and even so, the embodiment gap between human hand and robot hand may result different outcomes to the target object. In short, prior work implicitly assumes matching kinematical constraints or contact regions would result in same robot-object interaction, which may not always hold. Our work differs from prior work that we directly aim to reproduce the interaction without such assumptions. Specifically, we learn a retargeting model which outputs robot actions that can achieve same object trajectory given from human mocap data when executed.

Dexterous Manipulation. Several work have leveraged human mocap data for learning a dexterous robot hand policy, either using it as a target task demonstration [5], [9], or as a large dataset for extracting general prior [28], [29]. [5], [9] use human mocap data of a target task as a demonstration for training robot policy, while requiring finetuning with reinforcement-learning or teleoperation data. [28] use a perception module to extract human hand and finger poses from Internet videos [30], and learn a prior model which outputs plausible robot hand motions given visual input. [29] additionally extracts 2D contact locations and active object bounding box labels on top of human poses from human video dataset, which is used to train a visual encoder. The most similar setup to ours is [31], where the aim is to find a sequence of robot actions given human mocap data through optimizing a parametrized quasi-physical simulator. However, its main focus is within simulation, lacking real world evaluations. Additionally, as it aims to find a sequence of robot actions given a single sequence of human and object motion, retargeting process can be computationally expensive when given with a set of different motions of a single object being manipulated by human hand. Our work directly aims to perform dexterous manipulation in the real world, and shows generalization capabilities to unseen motion of the given object and human hand.

III. Method

We propose a learning-based framework that transforms human hand manipulation demonstrations into a sequence of robotic actions, enabling the robot hand to accurately imitate the same manipulation task. Specifically, our method takes as input a target object trajectory $\mathbf{O} \in \mathbb{R}^{o \times t}$ and a human hand demonstration $\mathbf{H} \in \mathbb{R}^{h \times t}$, and it outputs the corresponding plausible robot action trajectory $\mathbf{R} \in \mathbb{H}^{r \times t}$:

$$\mathbf{R} = \mathcal{F}(\mathbf{O}, \mathbf{H}) \tag{1}$$

, where t represents the number of timesteps, $o = 9$ (3 for position, 6 for rotation), $h = 81$ (3 for human wrist position, 6 for human wrist rotation, and $72 = 24 \times 3$ for each finger joint's 3D position w.r.t. human wrist frame), and $r = 25$ (3 for desired robot wrist position, 6 for desired robot wrist rotation, and 16 for desired robot hand joint angles), respectively. We use the 6D rotation representation [32].

Our framework $\mathcal{F}$ is built by learning the joint spatio-temporal manifold over $\mathbf{O}$, $\mathbf{H}$, and $\mathbf{R}$ by training a convolutional autoencoder model [33], [34]:

$$(\mathbf{O}, \mathbf{H}, \mathbf{R}) \approx \mathbf{\Psi}_{\text{dec}} \left(\mathbf{\Psi}_{\text{enc}} \left(\mathbf{O}, \mathbf{H}, \mathbf{R} \right) \right), \tag{2}$$

where $\mathbf{\Psi}_{\text{enc}}$ and $\mathbf{\Psi}_{\text{dec}}$ are the 1-D temporal convolution encoder and decoder applied to the concatenated triplet $(\mathbf{O}, \mathbf{H}, \mathbf{R})$. The encoded bottleneck layer $\mathbf{\Psi}_{\text{enc}} \left(\mathbf{O}, \mathbf{H}, \mathbf{R} \right) = \mathbf{L}$ represents the manifold latent code modeling the correlations among human hand, robot action, and target object trajectory during manipulation. This learned manifold enables us to estimate missing components through an optimization-based framework. For example, given human hand motion $\mathbf{H}$ and target object trajectory $\mathbf{O}$, we infer the corresponding plausible robot hand trajectory $\mathbf{R}$ by optimizing the latent

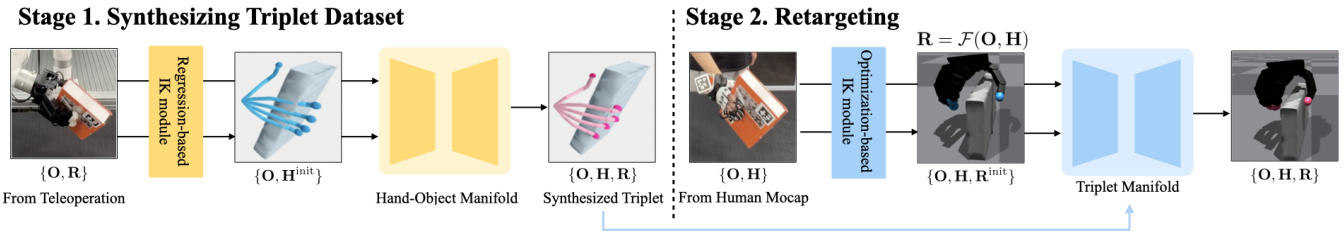

Fig. 2: **Overview of the Proposed Framework.** We first synthesize the paired triplet dataset consisting of robot action and human motion achieving the same object trajectory, followed by learning a retargeting module. The retargeting model is evaluated in real world, and we use IsaacGym simulator for visualization only.

code $\mathbf{L}$ as follows:

$$\mathbf{L}^* = \arg\min_{\mathbf{L}} \|\mathbf{\Psi}^{\mathbf{O}}_{\text{dec}}(\mathbf{L}) - \mathbf{O}\|_2, \tag{3}$$

where $\mathbf{\Psi}^{\mathbf{O}}_{\text{dec}}$ is the object trajectory component decoded from the manifold latent code $\mathbf{L}$, which is compared to the desired object trajectory $\mathbf{O}$. Once the optimal $\mathbf{L}^*$ is found, the desired robot hand motion $\mathbf{R}$ can be computed as:

$$\mathbf{R} = \mathbf{\Psi}^{\mathbf{R}}_{\text{dec}}(\mathbf{L}^*) \tag{4}$$

, by applying the decoder and extracting the robot hand component from the output, denoted as $\mathbf{\Psi}^{\mathbf{R}}_{\text{dec}}$. Since the latent code optimization Eq. 2 is performed via a gradient decent method, choosing a good initial $\mathbf{L}^{\text{init}}$ is important. To achieve this, we first estimate an initial robot hand motion $\mathbf{R}^{\text{init}}$ from human motion $\mathbf{H}$ using a conventional Inverse Kinematics (IK)-based optimization, by matching robot fingertip positions to human fingertip positions. We then use this as the input for the encoder to initialize the latent code: $\mathbf{L}^{\text{init}} = \mathbf{\Psi}_{\text{enc}}(\mathbf{O}, \mathbf{H}, \mathbf{R}^{\text{init}})$.

To learn the manifold space over $(\mathbf{O}, \mathbf{H}, \mathbf{R})$, it is necessary to collect paired data for the supervision, consisting of human and robot hand motions that result in the same object manipulation. However, it is infeasible to obtain such dataset, as both demonstrations cannot be performed simultaneously. Our key insight is to synthesize plausible pseudo-ground-truth pairs, which we describe next.

A. Synthesizing Pseudo-GT Triplet DB.

To learn the manifold space described in Eq. 2, we need a dataset containing a set of triplets $\{\mathbf{O}_i, \mathbf{H}_i, \mathbf{R}_i\}_{i=1}^N$. However, collecting such a dataset is impractical. Our solution is to synthesize corresponding human hand motion samples $\mathbf{H}_i$ based on a collected robot teleoperation data $\{\mathbf{O}_i, \mathbf{R}_i\}$.

Specifically, given a target object, we collect two separate datasets: via human mocap manipulation demonstrations $\{\mathbf{O}^M_j, \mathbf{H}^M_j\}_{j=1}^M$, and via teleoperation $\{\mathbf{O}_i, \mathbf{R}_i\}$. Then, we build a framework $\mathcal{E}$ to synthesize hand motion as follows:

$$\mathbf{H}_i = \mathcal{E}(\mathbf{O}_i, \mathbf{R}_i) \tag{5}$$

, where our framework $\mathcal{E}$ is composed of two modules: (1) a regressor Ω to estimate an initial human hand motion $\mathbf{H}^{\text{init}}_i = \Omega(\mathbf{R}_i)$, and (2) a manifold-based refinement process ψ to improve the $\mathbf{H}^{\text{init}}_i$ considering the object trajectory $\mathbf{O}_i$, achieved via manifold learning similar to Eq. 2. We describe each module below.

Learning The Manifold Space for $\{\mathbf{O}_i, \mathbf{H}_i\}$. We first learn the joint spatio-temporal manifold space over $\mathbf{O}$ and $\mathbf{H}$, similar to Eq. 2, by training an a temporal-convolutional autoencoder model [33], [34] using the human mocap manipulation dataset $\{\mathbf{O}^M_j, \mathbf{H}^M_j\}$:

$$(\mathbf{O}, \mathbf{H}) \approx \psi_{\text{dec}}\left(\psi_{\text{enc}}\left(\mathbf{O}, \mathbf{H}\right)\right) \tag{6}$$

, where the bottleneck latent code $\mathbf{l} = \psi_{\text{enc}}\left(\mathbf{O}, \mathbf{H}\right)$ captures the spatio-temporal correlation between object trajectory $\mathbf{O}$ and the corresponding hand motion $\mathbf{H}$. Once trained, this model can be used to infer the corresponding hand motion $\mathbf{H}_i$, given object trajectory $\mathbf{O}_i$ by finding the optimal latent code ${\mathbf{l}_i}^*$:

$${\mathbf{l}_i}^* = \arg\min_{\mathbf{l}} \|\psi^{\mathbf{O}}_{\text{dec}}(\mathbf{l}) - \mathbf{O}_i\|_2 \tag{7}$$

, where $\psi^{\mathbf{O}}_{\text{dec}}$ is the object trajectory component decoded from the manifold latent code $\mathbf{l}$. Once we obtain the optimal ${\mathbf{l}_i}^*$, the desired human hand motion $\mathbf{H}_i$ can be obtained as:

$$\mathbf{H}_i = \psi^{\mathbf{H}}_{\text{dec}}({\mathbf{l}_i}^*). \tag{8}$$

As in optimization for Eq. 2, selecting a good initial latent code $\mathbf{l}^{\text{init}}$ is important. Thus, we first estimate the initial hand motion ${\mathbf{H}_i}^{\text{init}}$ from robot action $\mathbf{R}_i$, and apply it to the pre-trained encoder ${\mathbf{l}_i}^{\text{init}} = \psi_{\text{enc}}(\mathbf{O}, {\mathbf{H}_i}^{\text{init}})$. However, unlike the previous case where a traditional IK solver is used, here we train a neural regressor to estimate ${\mathbf{H}_i}^{\text{init}}$.

Regressing Hand Motion from Robot Action. One way to estimate the initial human hand motion ${\mathbf{H}_i}^{\text{init}}$ from the provided robot hand $\mathbf{R}_i$ (obtained via teleoperation) is through a traditional Inverse Kinematics (IK), aligning corresponding fingertip and joint positions. However, we found this IK optimization from robot hand to human hand unstable unlike the opposite direction, human hand to robot hand, since the human hand typically has a higher degree of freedom with more fingers. As a solution, we build a neural regressor ${\mathbf{H}_i}^{\text{init}} = \Omega(\mathbf{R}_i)$, which we train on a dataset of paired human and robot hand motions $\{\mathbf{H}_k, {\mathbf{R}_k}^{\text{IK}}\}$. To build the paired DB, we compute the robot hand ${\mathbf{R}_k}^{\text{IK}}$ from human hand $\mathbf{H}_k$ using traditional IK optimization, minimizing the difference between human fingertip and robot fingertip positions. To represent the human hand and robot hand, we include the rotation and position of the wrist, and the position of the finger joint and fingertips (24 positions for the human hand and only 4 fingertips for the robot defined w.r.t. the wrist frame). Then, we train the neural network regressor to predict

human hand motion from robot action, $\mathbf{H}_i^{\text{init}} = \Omega(\mathbf{R}_i)$. Our regression model Ω consists of 6 layers of multi-head self-attention [35] with 8 heads, each with an embedding dimension of 256. Ω operates per frame, converting each robot hand configuration to human hand pose.

Hand Motion Refinement. While the initial estimate of human hand motion $\mathbf{H}_i^{\text{init}}$ produced by Ω shows a certain level of visual plausibility, the quality is limited due to the limited quality of the supervision derived from traditional IK, and, more importantly, its failure to account for interactions between hand and object trajectories, by taking only robot hand as the input. Our manifold-based optimization using the model of Eq 6 significantly enhances the quality of hand motion synthesis, by capturing the relationship between human hand motion and object movement, resulting in the final output $\mathbf{H}_i$.

While motion manifold autoencoder Eq 6 is trained with a fixed window size, it can be applied to arbitrary lengths of $\{O_i, H_i\}$ by applying it at each starting point in a sliding window fashion, and optimizing latent codes at all time window together via Eq 7, along with enforcing temporal consistency of the overlapped output.

B. Hardware System Setup for Data Collection

To collect the required human mocap demo and robot teleoperation data, we build a multi-camera system with 16 cameras paired with wearable motion capture devices and gloves to capture 3D human body motion and object movements, as shown in Fig. 3 following the system of [36]. The 3D human body and hand motions cues are obtained from wearable mocap devices. The multi-camera system is used to track the 3D object movement by tracking the attached aruco markers on the object and the gloves, as shown in Fig. 4, where the marker on the gloves are required to align the human motion and object in the same 3D coordinate system. Multi camera system is synced and spatially calibrated. In our setup, objects, human hands, and robot arms and hands are located in the common 3D coordinate, with 30Hz capture frequency. After calibration, we collect human mocap demo and robot teleoperation data using the same system. To perform robotic teleoperation, we directly use the teleoperator's wrist pose w.r.t. pelvis frame and hand joint angles acquired from the mocap device, as a robot action.

IV. Evaluations

A. Experimental Setup

We choose three objects and corresponding tasks with different characteristics to show the validity of our framework. Three objects and its corresponding tasks are as below.

Bottle. The robot must pick the bottle from a randomized starting location, and place it within the target location without tipping it over . The bottle's diameter is sufficiently large such that a human hand cannot fully encompass it, whereas the robot hand is capable if enveloping the whole bottle due to larger hand size. Naively matching robot and human hand fingertips may result an unstable grasp. **Bowl.** The robot must pick the bowl from a randomized starting location while maintaining upright rotation, and place it within the target location. The bowl's concave shape induces a complicated contact interaction between human hand and object hand, and a precise control is needed for the robot in order to pick up the bowl without tilting it. **Book.** The robot must pick up the book and reorient it in order to make it stand vertically. Due to the book's flatness and size, the robot hand must maintain a fine, consistent contact with the book during reorientation to prevent slipping. Additionally, careful placement is crucial to ensure the book remains upright.

Fig. 3: **System Overview:** Our system consists of 16 synchronized cameras, an xArm6 robot arm, and a 16-DoF Allegro robot hand.

Fig. 4: Objects used in the experiment and a marker system for 3D tracking.

For each task of bottle, bowl, book, we collect 113, 100, 114 human mocap demonstrations and 92, 114, 93 robot teleoperation demonstrations, respectively. There exists many different possible object trajectories achieving the same task, emphasizing the need of synthetic data generation pipeline we propose. We train $\mathbf{\Psi}_{enc,\,dec}$ and $\psi_{enc,\,dec}$ per each task, and a regressor Ω is shared across tasks. Each dataset is divided into trainset and validset with ratio 9 : 1.

We use position control for controlling the robot arm and robot hand. Note that the desired wrist pose and hand joint values (i.e. robot actions that are actually fed as commands) may differ from actual values based on the interaction between robot and the object, as shown in Fig. 5. While the

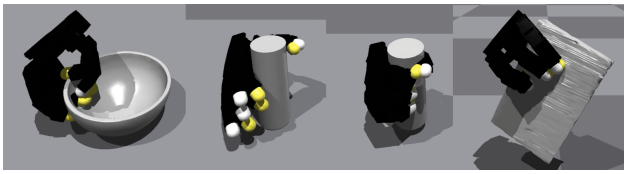

Fig. 5: **Visualization of robot teleoperation dataset**. **Yellow**: desired robot joint values. **White**: actual robot joint values. The dataset is collected in real world, and Isaac Gym simulator is only used for rendering.

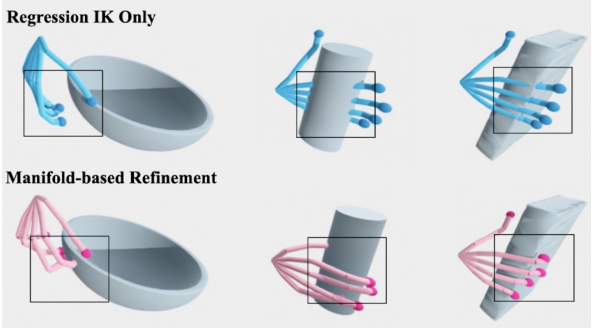

Fig. 6: Qualitative comparison before and after applying hand motion refinement model for synthetic data generation. **Blue hand** and **red hand** indicates before and after refinement, respectively. Hand motion after refinement shows more plausible hand movements.

Task	Method	Penetration (m, ↓) tip / tip+mid	Contact(m, ↓) tip / tip+mid
Bottle	Ours w/o refine.	0.003 / **0.002**	0.014 / 0.014
	Ours	**0.002 / 0.002**	**0.010 / 0.009**
Bowl	Ours w/o refine.	**0.005 / 0.003**	0.013 / 0.013
	Ours	0.009 / 0.007	**0.010 / 0.010**
Book	Ours w/o refine.	0.012 / 0.006	0.037 / 0.034
	Ours	**0.004 / 0.004**	**0.024 / 0.026**
Total	Ours w/o refine.	0.006 / **0.004**	0.021 / 0.020
	Ours	**0.005 / 0.004**	**0.014 / 0.014**

TABLE I: Comparison between ours and without refinement. tip indicates evaluating fingertip, and tip+mid indicates evaluating both fingertip and mid joint.

visualization of desired robot proprioception penetrates the surface, the actual robot is in contact with the object along its surface. Such discrepancy makes kinematics-based retargeting methods(i.e. matching human and robot fingertips) suboptimal, as the robot finger may slip due to inadequate contact forces.

We aim to the answer the following questions through the experiment. **(Q1)** Does a hand motion refinement process along manifold of human mocap produces more physically plausible and natural human motion for synthetic human-object interaction data? **(Q2)** Is our regression model Ω necessary to provide a good latent initialization for hand motion refinement? **(Q3)** Can our retargeting model $\mathcal{F}$ better translate human mocap data to robot action data compared to baselines? **(Q4)** Can our retargeting model $\mathcal{F}$ generalize to unseen object trajectories? **(Q5)** Is our retargeting model robust to noise in the human mocap demo?

B. Synthetic Paired Dataset Generation Model $\mathcal{S}$

In this section, we verify our design choice for synthetic dataset generation pipeline.

Effectiveness of human hand motion refinement. First, we evaluate the performance of human hand motion refinement, based on the following metrics.

Contact. Measured by the distance between human fingertips or finger middle joints and closest object surface during manipulation. Middle joint refers to joint between distal and middle bone. Within all our tasks, human's all fingertips and middle joints are naturally in contact with the target object during manipulation.

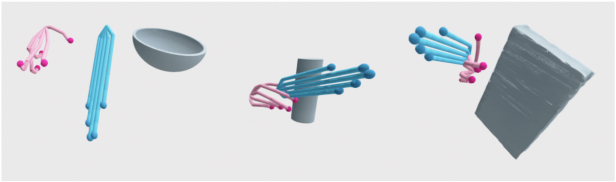

Fig. 7: Ablation on different initial estimate before refinement. **Blue hand** is the initial estimate, which uses robot wrist pose and zero-pose human fingers instead of our regression model. **Red hand** is the refined hand motion.

Task	Method	COM Error (m, ↓)	Ori Error (rad, ↓)
Bottle	Fingertip	0.097	0.31
	Fingertip + Midjoint	0.085	**0.15**
	Ours	**0.050**	0.22
Bowl	Fingertip	**0.051(0.052)**	0.68(0.72)
	Fingertip + Midjoint	0.058	0.72
	Ours	0.054(0.054)	**0.57(0.49)**
Book	Fingertip	0.055	0.24
	Fingertip + Midjoint	0.062	0.25
	Ours	**0.052**	**0.22**
Total	Fingertip	0.067	0.42
	Fingertip + Midjoint	0.068	0.39
	Ours	**0.052**	**0.34**

TABLE II: Error metrics for our method and baselines. For the Bowl task, we also compare the robustness of our method with the Fingertip baseline with noisy input. Values in parentheses indicate when the human hand motion is from noised mocap.

Penetration. Measured by the penetration depth of human fingertips or finger middle joints. A plausible human hand motion should not have penetration with the object.

All metrics are computed on the synthetic dataset we generated, $\{O_i, H_i^{init}\}$ and $\{O_i, H_i\}$, each referring to before and applying hand motion refinement. O_i is from teleoperation dataset. Metric is averaged over all frames within the duration when the object is in motion, based on the generated synthetic hand motion and the ground truth object trajectory.

Results are in Table. I. Applying motion editing produces more natural and physically plausible human hand motions, reducing both contact and penetration errors in most cases. Fig. 6 shows qualitative examples before and after applying motion editing. The motion editing model successfully corrects wrong finger positions(i.e. fingertip not contacting the bowl, fingers penetrating the book) and wrong wrist pose(i.e. human wrist rotation is inaccurate making the whole hand to penetrate the bottle).

Necessity of regression-based IK model. To demonstrate the effectiveness of initializing based on regression-based IK model, we compare an alternative initialization which uses robot wrist pose and zero-pose human fingertip positions w.r.t. human wrist frame. As shown in Fig. 7, as zero hand initial pose are far away from plausible hand motion, latent optimization becomes unstable and generates unnatural hand and object interaction.

C. Human-to-Robot Retargeting Model $\mathcal{F}$

In this section, we evaluate our Retargeting Model $\mathcal{F}$ within the aforementioned three tasks, with the following metrics.

COM and Ori Error: These two terms measure the $L2$ norm between the object's target trajectory and the measured trajectory using our retargeting model, as in [37]. We compute the error within succeeded trajectories, as failure trajectories may show random object motion or no object motion, which makes error term less relevant to the model's performance (e.g. it is hard to compare object not moving at all vs. object being tipped over during grasping.).

Success Rate and Number of Completed Subtasks. For each task, we define task-specific metric and success criteria. The sub tasks are divided into pick and place, where 'pick' is determined by whether the robot successfully lifts the object(with all points off the ground), and 'place' is determined by whether the robot successfully places the object while maintaining balance. We evaluate as success when both the pick and place tasks were completed. For bowl task, we have an additional constraint to rotation: if the bowl tilts more than 45 degrees in the direction of the object's z-axis, it is considered a failure.

Baselines. We consider two types of baselines which are commonly used when learning robot policy from human mocap data or when performing robotic teleoperation. **Fingertip Matching**, which uses optimization-based IK to directly match robot hand fingertips to human hand fingertips, and **Fingertip and Middle Joint Matching**, which has additional constraint which enforces middle joints of robot hand and human hand to be in an identical position. Baselines do not take the interacting object into account, but rather considers the human hand's kinematic information only.

Generalization capability. To check whether our retargeting model generalizes to unseen human hand motion and object trajectory, we do the following evaluation. First, we split the human mocap data $\{O, H\}$ into train dataset and validation dataset, and use trainset only for human hand refinement model training. Then, the validset is used for evaluating the retargeting model $\mathcal{F}$. Note that we do not assume any paired ground truth robot action for both trainset and validset(i.e. both sets can be unseen to our retargeting model), but rather utilize the synthetic data we generated.

Table. II and Table III show the evaluated results of our model and baselines. Each task was evaluated over 10 episodes for the Bottle and Bowl task, and 11 episodes for the Book task. Overall, our method consistently achieved lower COM and Ori errors, resulting in a significantly lower total loss compared to the baselines. Our model also achieved higher success rate and sub task completion. While the baseline methods occasionally outperformed ours in either accuracy or task success, there was no instance where they demonstrated superior performance in both aspects simultaneously. Considering both aspects together and regarding overall robustness, our method outperforms.

Robustness to noisy mocap demo. While our system utilizes a multi-view camera system for accurate mocap, it is not always a viable option for collecting human mocap data. A promising alternative is to use vision-based human-objects reconstruction models. However, these models are often inaccurate, producing noisy human hand and finger poses.

Task	Method	# Subtasks	Success
Bottle	Fingertip	1.0	0.2
	Fingertip + Midjoint	0.6	0.2
	Ours	**1.7**	**0.7**
Bowl	Fingertip	1.4(1.1)	0.5(0.3)
	Fingertip + Midjoint	1.0	0.2
	Ours	**1.8**(**1.9**)	**0.8**(**0.9**)
Book	Fingertip	**1.36**	**0.45**
	Fingertip + Midjoint	1.36	0.36
	Ours	1.08	0.27
Total	Fingertip	1.26	0.39
	Fingertip + Midjoint	0.99	0.25
	Ours	**1.54**	**0.59**

TABLE III: Number of completed subtasks and success rate for our method and baselines. Values in parentheses indicate when the human hand motion is from noised mocap.

As we aim to develop a framework that can scale robotic data by retargeting human motions, we evaluate our model's robustness with noisy mocap data. Specifically, we choose Bowl task and Fingertip as baseline to compare against our method, as it has best performance among baselines and tasks. We add a gaussian noise to the validset we used for evaluation above with mean=0.001 at all dimension of human hand, from wrist pose to finger positions. The results are in Table. II and Table. III, where the values in parentheses indicate evaluation under noisy mocap. Surprisingly, although we add a very small noise, the performance of Fingertip baseline drops from 0.5 to 0.3 in terms of success rate, while ours got even better, from 0.8 to 0.9 success rate. COM and Ori Error stays similar, as it is computed over succeeded trajectories. This intuitively shows the instability of kinematics based baseline, which assumes mocap to be accurate. As our model learns the manifold of human hand and object interaction itself(i.e., the model implicitly knows robot hand or human hand should be in contact with the object when it is moving.), it shows robustness to certain level of noise.

V. Discussion and Limitations

In this work, we developed a framework for learning a retargeting model which translates human mocap demo to a sequence of plausible robot actions for reproducing the manipulation. Under a carefully designed pipeline, we achieve superior performance to baselines in multiple real world dexterous manipulation tasks, even within noisy mocap data. While our framework showed generalization capabilities along different trajectories within the same object, we have separate models for each object. Moreover, we only use object pose to represent each object. Learning a general, unified retargeting model along with rich representation induced from mocap data (i.e., proximity between hand and object) with a more scaled experiment will be an interesting future direction to pursue.

ACKNOWLEDGMENT

This work was supported by NRF grant funded by the Korean government (MSIT) (No. 2022R1A2C2092724), and IITP grant funded by the Korean government (MSIT) [RS-2021-II211343, AI Graduate School Program (SNU)]. H. Joo is the corresponding author.

REFERENCES

[1] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020.

[2] Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. In *RSS*, 2022.

[3] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *RSS*, 2023.

[4] Sridhar Pandian Arunachalam, Sneha Silwal, Ben Evans, and Lerrel Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. In *ICRA*, 2023.

[5] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *ECCV*, 2022.

[6] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *CVPR*, 2024.

[7] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *CVPR*, 2021.

[8] Jiye Lee and Hanbyul Joo. Mocap everyone everywhere: Lightweight motion capture with smartwatches and a head-mounted camera. In *CVPR*, 2024.

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

[10] Shivin Dass, Wensi Ai, Yuqian Jiang, Samik Singh, Jiaheng Hu, Ruohan Zhang, Peter Stone, Ben Abbatematteo, and Roberto Martin-Martin. Telemoma: A modular and versatile teleoperation system for mobile manipulation. *arXiv preprint arXiv:2403.07869*, 2024.

[11] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: teleoperation with immersive active visual feedback. In *CoRL*, 2024.

[12] Shiqi Yang, Minghuan Liu, Yuzhe Qin, Runyu Ding, Jialong Li, Xuxin Cheng, Ruihan Yang, Sha Yi, and Xiaolong Wang. Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation. In *CoRL*, 2024.

[13] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[14] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *CoRL*, 2024.

[15] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *CoRL*, 2024.

[16] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IROS*, 2024.

[17] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *CoRL*, 2024.

[18] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *ICRA*, 2023.

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

[20] Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. Ace: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In *SIGGRAPH Asia*, 2023.

[21] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.

[22] Sunwoo Kim, Maks Sorokin, Jehee Lee, and Sehoon Ha. Human-conquad: human motion control of quadrupedal robots using deep reinforcement learning. In *SIGGRAPH Asia*, 2022.

[23] Tianyu Li, Hyunyoung Jung, Matthew Gombolay, Yong Kwon Cho, and Sehoon Ha. Crossloco: Human motion driven control of legged robots via guided unsupervised reinforcement learning. In *ICLR*, 2024.

[24] Albert Wu, Ruocheng Wang, Sirui Chen, Clemens Eppner, and C Karen Liu. One-shot transfer of long-horizon extrinsic manipulation through contact retargeting. In *IROS*, 2024.

[25] Yuming Du, Philippe Weinzaepfel, Vincent Lepetit, and Romain Brégier. Multi-finger grasping like humans. In *IROS*, 2022.

[26] Arjun Lakshmipathy, Dominik Bauer, Cornelia Bauer, and Nancy S Pollard. Contact transfer: A direct, user-driven method for human to robot transfer of grasps and manipulations. In *ICRA*, 2022.

[27] Arjun S Lakshmipathy, Jessica K Hodgins, and Nancy S Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations. *arXiv preprint arXiv:2402.04820*, 2024.

[28] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In *CoRL*, 2023.

[29] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. In *RSS*, 2024.

[30] 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 *ECCV*, 2018.

[31] Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Quasisim: Parameterized quasi-physical simulators for dexterous manipulations transfer. In *ECCV*, 2024.

[32] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

[33] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. In *ACM Transa. Graph.*, 2016.

[34] Jiye Lee and Hanbyul Joo. Locomotion-action-manipulation: Synthesizing human-scene interactions in complex 3d environments. In *ICCV*, 2023.

[35] A Vaswani. Attention is all you need. In *Neurips*, 2017.

[36] Jeonghwan Kim, Jisoo Kim, Jeonghyeon Na, and Hanbyul Joo. Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. *arXiv preprint arXiv:2401.10232*, 2024.

[37] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *ICRA*, 2023.