2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

One-Policy-Fits-All: Geometry-Aware Action Latents for Cross-Embodiment Manipulation

Juncheng Mu[1,2*] Sizhe Yang[1,3*] Hojin Bae[2*] Feiyu Jia[1,4]
Qingwei Ben[1,3] Boyi Li[5†] Huazhe Xu[2†] Jiangmiao Pang[1†]
[1]Shanghai AI Laboratory [2]Tsinghua University [3]The Chinese University of Hong Kong
[4]University of Science and Technology of China [5]NVIDIA
Project page: https://mujc2021.github.io/opfa/

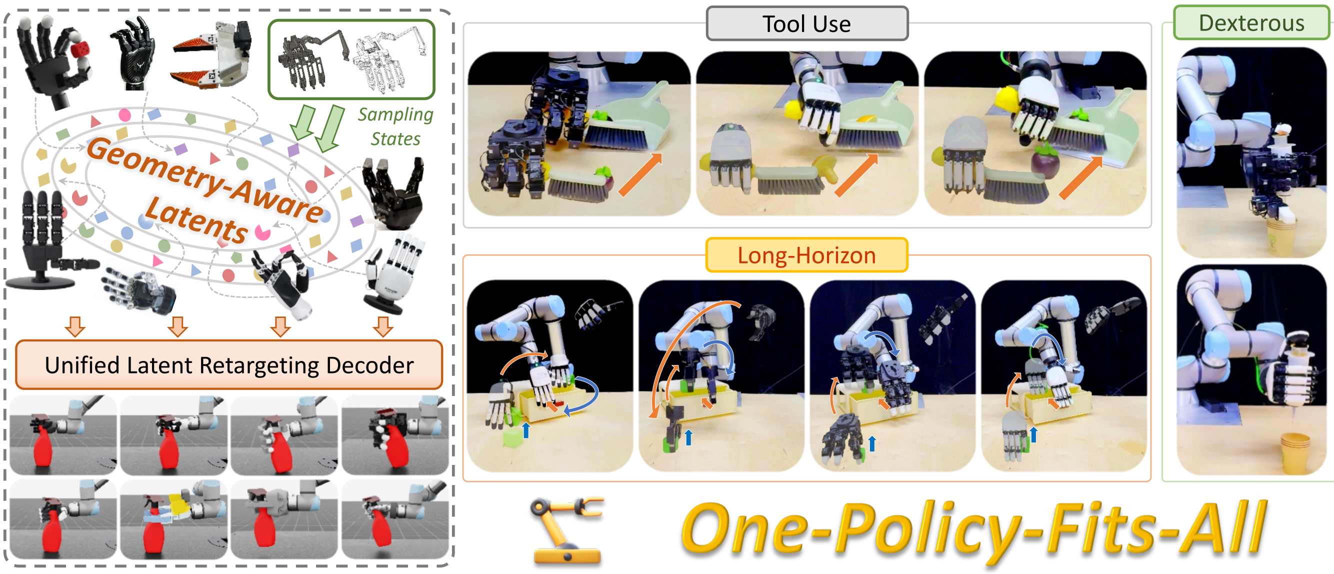

Fig. 1: We introduce **One-Policy-Fits-All** (OPFA), a general framework for cross-embodiment manipulation. OPFA leverages the geometric structures of diverse end-effectors to construct a unified latent action representation, and employs a unified latent retargeting decoder to recover embodiment-specific actions. This design enables seamless skill transfer across grippers and dexterous hands, offering a scalable solution to data scarcity and enabling rapid adaptation to new embodiments.

***Abstract*— Cross-embodiment manipulation is crucial for enhancing the scalability of robot manipulation and reducing the high cost of data collection. However, the significant differences between embodiments, such as variations in action spaces and structural disparities, pose challenges for joint training across multiple sources of data. To address this, we propose *One-Policy-Fits-All* (OPFA), a framework that enables learning a single, versatile policy across multiple embodiments. We first learn a *Geometry-Aware Latent Representation* (GaLR), which leverages 3D convolution networks and transformers to build a shared latent action space across different embodiments. Then we design a unified latent retargeting decoder that extracts embodiment-specific actions from the latent representations, without any embodiment-specific decoder tuning. OPFA enables end-to-end co-training of data from diverse embodiments, including various grippers and dexterous hands with arbitrary degrees of freedom, significantly improving data efficiency and reducing the cost of skill transfer. We conduct extensive experiments across 11 different end-effectors. The results demonstrate that OPFA significantly improves policy performance in diverse settings by leveraging heterogeneous embodiment data. For instance, cross-embodiment co-training can improve success rates by more than 50% compared to single-source training. Moreover, by adding only a few demonstrations from a new embodiment (e.g., eight), OPFA can achieve performance comparable to that of a well-trained model with 72 demonstrations.**

I. INTRODUCTION

Imitation learning [1]–[3] has emerged as a pivotal paradigm in robotics, offering learning-based solutions for complex real-world manipulation tasks. However, unlike learning in text and vision domains [4]–[6] where data typically comes from a single modality, robotic manipulation is inherently coupled with the physical characteristics of the end-effector, posing two key challenges for policy generalization: (i) end-effectors vary drastically in morphology and degrees of freedom—from simple grippers to highly articulated anthropomorphic hands, resulting in fundamentally different action spaces that hinder the joint exploitation of cross-embodiment data; (ii) introducing a new end-effector typically requires collecting a large amount of embodiment-specific data, which is both costly and limits scalability.

Existing methods struggle to fully exploit cross-embodiment data due to two main challenges: (i) substantial differences in action dimensions and spaces across embodiments, and (ii) large structural discrepancies that create a pronounced embodiment gap. If a unified policy could co-train cross-embodiment data without conflicts, the problem of data scarcity would be effectively alleviated.

Furthermore, decoupling policy learning from a specific end-effector would enable more versatile and generalizable policies, as well as more data-efficient learning, thereby greatly facilitating practical real-world deployment.

Recently, several approaches [7]–[12] have explored cross-embodiment co-training to expand data scale and improve generalization across embodiments. Most of these methods target settings where grippers act as the end-effector, achieving cross-embodiment capability at the arm–gripper level either by unifying the action space [7] or by employing embodiment-specific decoders [11]. However, extending such strategies to dexterous hands—with their higher degrees of freedom and more complex physical structures—remains substantially more challenging. Naively assigning a separate action head or decoder to each embodiment [12], [13] neglects the geometric structure of the hand, while restricting each decoder to single-source data, leading to low data efficiency and limited generalization.

To overcome the limitations of prior methods and enable cross-embodiment co-training across a wide range of end-effectors, particularly dexterous hands, we propose One-Policy-Fits-All (OPFA), a general framework for cross-embodiment manipulation. Specifically, OPFA first learns a geometry-aware action latent representation (GaLR), which captures the spatial structure of an end-effector's reachable states from point clouds (derived directly from joint angles) using 3D convolutional networks [14] and transformers [15]. The GaLR produced by the encoder unifies action space dimensions across multiple embodiments while consistently encoding geometric information. This transforms action prediction into a latent-space prediction of spatial structures across different embodiments. During training, reachable-state point clouds from multiple embodiments are sampled, and the decoder recovers embodiment-specific joint angles from the encoded GaLR. This enables the encoder and decoder to be trained jointly in an end-to-end manner without any additional manual annotation. It is noteworthy that, unlike prior approaches [11]–[13] that train separate decoders for each embodiment, we design a unified latent retargeting decoder capable of handling diverse embodiments. Finally, the trained spatial encoder, decoder, and the constructed GaLR can be integrated into diverse action prediction methods (e.g., DP [16], DP3 [17]), effectively transferring action prediction to the latent space. In this way, data from multiple embodiments are represented in a unified dimensionality while sharing geometric information, substantially enhancing generalization across diverse end-effectors.

Extensive experiments in both simulation and the real world validate the strong performance of OPFA. We evaluate it on 11 diverse embodiments—ranging from two-finger grippers (UMI [18], Robotiq-2F), to three-finger (Robotiq-3F), four-finger (Leap [19], Allegro), and five-finger hands (Inspire Hand [20], XHand [21], etc.)—across 14 challenging manipulation tasks such as pouring and sweeping. Comprehensive cross-embodiment evaluations show that OPFA consistently outperforms both single-embodiment training and naive cross-embodiment co-training with embodiment-specific decoders. Notably, OPFA can achieve success rates on a new end-effector comparable to those of a well-trained model (72 demonstrations), while requiring as few as eight demonstrations. In summary, our contributions are three-fold:

- We construct a Geometry-Aware Latent Representation (GaLR) that aligns the action space dimensions across different end-effectors and learns geometric information without any manual annotation cost.
- We propose an end-to-end cross-embodiment co-training method that does not require any embodiment-specific decoder tuning.
- We conduct extensive experiments in both simulation and the real world, covering 11 different end-effectors, and demonstrate that OPFA significantly outperforms both self-training and naive co-training methods in various cross-embodiment settings.

II. RELATED WORKS

A. Cross-Embodiment Learning for Robot Arms

Recent efforts toward a universal policy have focused on cross-embodiment learning, with significant advancements for robot arms equipped with parallel-jaw grippers. These policies are often trained on large-scale datasets [22]–[30]. To handle the data heterogeneity across different robots, a common strategy is to unify the action space around the end-effector. Models like RT-1 [31] and RT-2 [32] predict end-effector motions and gripper states from visual and language inputs. Octo [11] utilizes a standardized end-effector space for pre-training and is uniquely designed to be adapted to new action spaces via modular adapters during fine-tuning. A more comprehensive approach is taken by RDT-1B [7], which standardizes all data into the *Physically Interpretable Unified Action Space*. Another effective paradigm involves learning a shared latent space to bridge morphological gaps between different arms [33]. These methods have proven effective for robot arms, which typically use simple parallel-jaw grippers. However, they do not directly address the distinct challenges posed by dexterous hands, which feature complex geometries and high degrees of freedom.

B. Cross-Embodiment Learning for Dexterous Hands

Cross-embodiment learning for multifingered robotic hands [34], [35] is more challenging due to their high dimensionality and significant structural diversity. A common strategy involves training embodiment-specific decoders [12], [13]. However, these approaches still restrict each decoder to training on data from a single embodiment. Other works leverage human motion as a prior via retargeting. Bauer et al. [13] trains a diffusion policy in a shared latent action space, with MANO-based [36] latent feature aligning. VideoDex [37] extracts actions directly from human videos. CrossDex [38] learns a universal policy in an abstract action space based on MANO model [36], which is then retargeted to robot-specific commands. However, due to the substantial structural discrepancies across different end-effectors, naively aligning them to a human-hand model may induce action conflicts. Instead of relying on pre-defined human

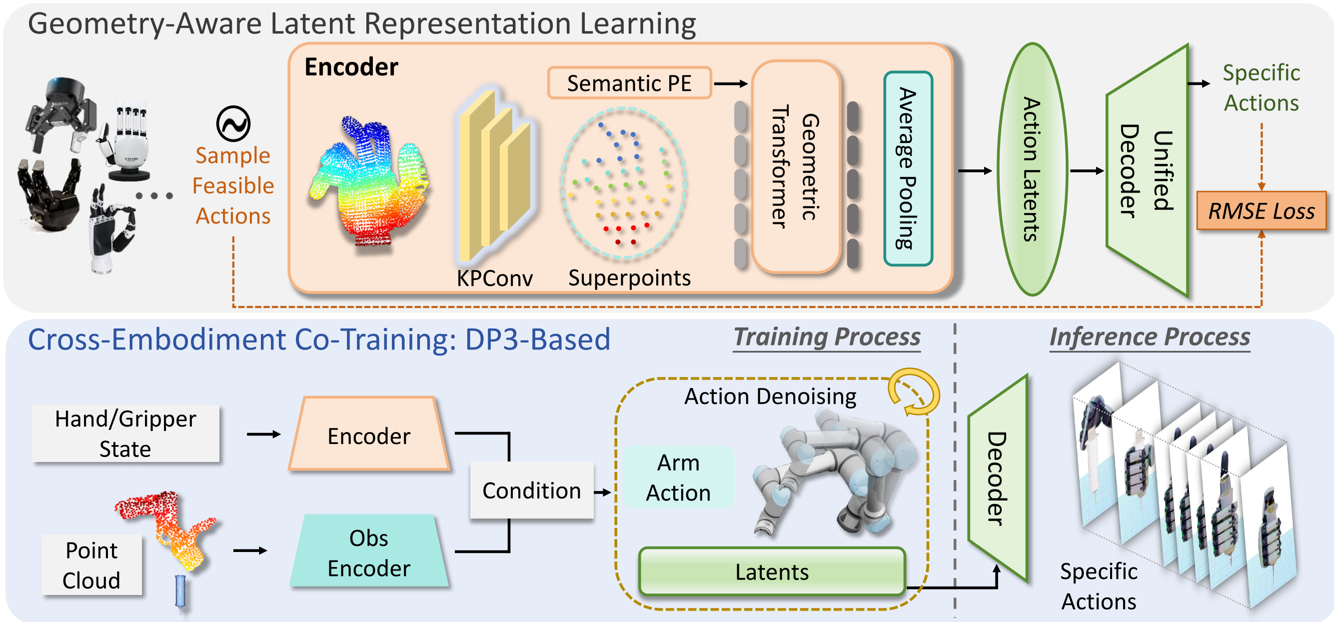

Fig. 2: The training pipeline of OPFA follows a two-stage paradigm. (1) We first construct a *Geometry-Aware Latent Representation* (GaLR) by encoding sampled reachable-state point clouds with 3D convolutions and geometric transformers for local/global feature extraction. A unified latent retargeting decoder then disentangles embodiment-specific actions from the latent space, enabling end-to-end training without manual annotations. (2) The pretrained encoder–decoder pair is integrated into any downstream policy (e.g., DP3), allowing cross-embodiment data to be jointly trained in a unified latent action space.

hand priors, OPFA directly learns a Geometry-Aware Latent Representation based on the sampled reachable-state point cloud of all the end-effectors to avoid action conflicts, and utilizes a unified decoder based on latent retargeting without any embodiment-specific decoder tuning.

III. METHOD

We propose *One-Policy-Fits-All* (OPFA), a general framework for cross-embodiment manipulation. OPFA adopts a two-stage paradigm: (i) a learning-based but annotation-free stage that learns a *Geometry-Aware Latent Representation* (GaLR), unifying the action spaces of diverse end-effectors; and (ii) a policy integration stage, where GaLR is embedded into the state condition and action prediction head of the policy (e.g., DP3 [17]), enabling end-to-end cross-embodiment co-training. The overall pipeline is shown in Figure 2.

A. Overview

We denote the set of embodiments as $\mathcal{M} = \{1, 2, \ldots, M\}$, where each embodiment $m \in \mathcal{M}$ is associated with a dataset $\mathcal{D}_m = \{\tau_i^m\}_{i=1}^{N_m}$, consisting of collected trajectories. Each trajectory is defined as $\tau_i^m = \left\{\left(\mathbf{o}_t^m, \mathbf{a}_t^m\right)\right\}_{t=1}^{T_i^m}$, where $\mathbf{o}_t^m$ denotes the observation at timestep t, $\mathbf{a}_t^m \in \mathbb{R}^{d_m}$ is the corresponding action, and T_i^m is the length of the trajectory. To fully leverage data from multiple embodiments, we aim to co-train $\{\mathcal{D}_m\}_{m=1}^{M}$. However, since d_m differs for each embodiment, unifying the action space presents a major challenge. Prior works [13], [39] often address this issue by applying embodiment-specific transformer heads or action decoders. However, for embodiment m, the embodiment-specific decoder can only be trained with data from $\mathcal{D}_m$. This limitation becomes particularly severe in few-shot learning scenarios, where data scarcity causes decoders to overfit and hinders effective skill transfer (see Section IV-C).

B. Learning Geometry-Aware Latent Representation

The construction of GaLR adopts an end-to-end encoder–decoder training scheme, with all training data generated automatically, entirely *without manual annotation*. For each embodiment $m \in \mathcal{M}$, we first sample a set of reachable states J^m, and apply the forward kinematics function f_{FK}^m and point sampling to obtain a set of training data $\{(\mathbf{a^m}, \mathcal{P})\}^m$, where $\mathbf{a^m} \in J^m$ is the joint angles used for supervision, and $\mathcal{P} \in \mathbb{R}^{|\mathcal{P}|\times 3}$ is the sampled point cloud. We apply a vision encoder f_θ to extract shared geometric features across multiple embodiments and states, mapping each $\mathcal{P}$ into the latent space: $\mathbf{z} = f_\theta(\mathcal{P}) \in \mathbb{R}^{d_{latent}}$. A unified decoder g_ψ then predicts the embodiment-specific joint angles $\hat{\mathbf{a}}^\mathbf{m} \in \mathbb{R}^{d_m}$ from the latent vector: $\hat{\mathbf{a}}^\mathbf{m} = g_\psi(\mathbf{z})$.

Because the number of dense points is relatively large, directly extracting features at this level results in redundancy and low computational efficiency. To address this, we perform three stages of downsampling to obtain multi-scale spatial features. Specifically, we denote the original dense point cloud as $\mathcal{P} \in \mathbb{R}^{|\mathcal{P}|\times 3}$, the first-level downsampled points as $\tilde{\mathcal{P}} \in \mathbb{R}^{|\tilde{\mathcal{P}}|\times 3}$, and the final downsampled points (*superpoints*) as $\hat{\mathcal{P}} \in \mathbb{R}^{|\hat{\mathcal{P}}|\times 3}$, where $|\hat{\mathcal{P}}| < |\tilde{\mathcal{P}}| < |\mathcal{P}|$. We then apply a geometric transformer [15] at the superpoint level $\hat{\mathcal{P}}$ to capture global gesture-aware representations.

Multi-scale local structure encoding. We perform multi-scale feature extraction on the downsampled point clouds with the 3D convolution [14]. Let $\mathcal{B}_r^3 = \{\mathbf{y} \in \mathbb{R}^3 \mid \|\mathbf{y}\| \le r\}$ be a ball of radius r centered at a query point $\mathbf{x}$. For a neighboring point $\mathbf{x_i}$, let $\mathbf{y_i} = \mathbf{x_i} - \mathbf{x} \in \mathcal{B}_r^3$ denote the relative coordinate of $\mathbf{x_i}$. The kernel points are defined as $\{\tilde{\mathbf{x}}_\mathbf{k} \mid k < K\} \subset \mathcal{B}_r^3$, and each kernel point $\tilde{\mathbf{x}}_\mathbf{k}$ is associated with a weight matrix $\mathbf{W}_k \in \mathbb{R}^{D_{\text{in}} \times D_{\text{out}}}$, where K is the number of kernel points, and $D_{\text{in}}, D_{\text{out}}$ are the input and output feature dimensions.

For a query point $\mathbf{x}$ with neighborhood $\{\mathbf{x_i}\}$ and input features $f_{\mathbf{x_i}} \in \mathbb{R}^{D_{\text{in}}}$, the convolution output is given by

$$g(\mathbf{x}) = \sum_i \sum_{k<K} h(\mathbf{y_i}, \tilde{\mathbf{x}}_\mathbf{k})\, \mathbf{W}_k f_{\mathbf{x_i}}, \tag{1}$$

where h measures the proximity between $\mathbf{y_i}$ and $\tilde{\mathbf{x}}_\mathbf{k}$. h is implemented as a linear decay function truncated at σ:

$$h(\mathbf{y_i}, \tilde{\mathbf{x}}_\mathbf{k}) = \max\left(0, 1 - \frac{\|\mathbf{y_i} - \tilde{\mathbf{x}}_\mathbf{k}\|}{\sigma}\right). \tag{2}$$

Here, σ is a hyperparameter that controls the influence radius of each kernel point.

Global gesture perception. The 3D convolution stage effectively extracts multi-scale local geometric features of the end-effectors and condenses them into a set of superpoints. Subsequently, to capture the global gesture information of each embodiment, we apply a geometric transformer [15] that performs cross-attention over the superpoints. Directly applying cross-attention over superpoints may lead to positional ambiguity. To address this issue, we utilize two positional embeddings. The first one is the naive coordinate positional embedding r^p. Additionally, we design a semantic positional embedding r^s, which provides a unified structural encoding across different embodiments. This embedding incorporates spatial semantics to preserve fine-grained positional information, thereby enabling more accurate and embodiment-agnostic geometry reasoning.

To be specific, we assign each superpoint $p \in \hat{\mathcal{P}}$ a semantic 2-dimensional index $\pi(p) = (u_p, v_p)$, where $u_p \in \{0, \ldots, 5\}$ denotes the finger-level index (palm: 0, thumb: 1, …, little finger: 5) and $v_p \in \mathbb{Z}_{\geq 0}$ denotes the segment-level index along the corresponding finger (each finger's segment levels start from 0). We form the 2D index vector $\mathbf{s}_p = [u_p, v_p]^\top \in \mathbb{R}^2$ and map it into the feature space: $r^s = \mathbf{s}_p \mathbf{W}^S \in \mathbb{R}^{d_t}$, where d_t is the superpoint feature dimension, and $\mathbf{W}^S \in \mathbb{R}^{2 \times d_t}$ is the projection matrices for semantic embedding.

Then the positional embedding r^s and r^p are added together, and fed into the geometric transformer:

$$\tilde{f}_p = \text{Transformer}\big(f_p, r^p + r^s\big), \quad p \in \hat{\mathcal{P}}. \tag{3}$$

Finally, we aggregate the refined superpoint features $\{\tilde{f}_p\}_{p \in \hat{\mathcal{P}}}$ via global average pooling to obtain the GaLR $z \in \mathbb{R}^{d_{\text{latent}}}$:

$$z = \frac{1}{|\hat{\mathcal{P}}|} \sum_{p \in \hat{\mathcal{P}}} \tilde{f}_p. \tag{4}$$

Unified decoder with latent retargeting. Although GaLR serves as a latent action representation to unify the action dimensions, embodiment-specific inference still requires recovering the actual joint angles. Prior approaches [13] typically train separate action decoders for each embodiment, which restricts each decoder to learning only from its own dataset, preventing effective utilization of shared information, and more importantly, may cause overfitting in few-shot learning scenarios (see Section IV-C).

To address this, we design a unified decoder g_ψ that predicts all joints $\hat{\Theta}$ in a hypothetical universal hand model $\mathcal{H}$, which contains every physically meaningful joint across all end-effectors (e.g., thumb yaw, thumb base flexion, index finger yaw, etc.). For each specific hand m, only the joints that exist on that hand are selected from $\mathcal{H}$, denoted as $\hat{\Theta}^m$. Finally, we can calculate the RMSE loss of the joint angles $\hat{\Theta}^m$ recovered from GaLR and the ground-truth joint angles $\mathbf{a^m}$, allowing the entire GaLR training framework to be optimized in an end-to-end manner.

C. Cross-Embodiment Policy Co-Training

By introducing GaLR as the general latent action representation, we can unify the training and inference process of different end-effectors. Given the cross-embodiment dataset $\{\mathcal{D}_m\}_{m=1}^M$, our objective is to learn a unified visuomotor policy $\pi : \mathcal{O} \to \mathcal{A}$, that maps visual observations $\mathbf{o} \in \mathcal{O}$ into actions $\mathbf{a} \in \mathcal{A}$ across different embodiments. We denote the function that generates GaLR from hand-specific actions (joint angles) as:

$$\mathcal{G} = f_\theta \circ f_{FK}^m. \tag{5}$$

Then, for each embodiment $m \in \mathcal{M}$, the latent policy specializes into

$$\pi_m : \mathbf{o}_t^m \mapsto \mathcal{G}(\mathbf{a}_t^m) \in \mathbb{R}^{d_{latent}}. \tag{6}$$

In this way, the policy π learns embodiment-agnostic visuomotor skills from the joint dataset $\{\mathcal{D}_m\}_{m=1}^M$ through GaLR. OPFA is base-policy-agnostic and can be seamlessly integrated into various policy architectures such as ACT [40] or DP3 [17]. The only modification required is to transform the action prediction process into GaLR prediction, while replacing the state terms $\mathbf{s_t^m}$ in the observation with the corresponding GaLR representation $\mathcal{G}(\mathbf{s_t^m})$. In our implementation, we adopt DP3 as the underlying policy backbone.

During training, the denoising process is applied to GaLR in the latent space. While during inference, we directly use the DP3 model trained on $\{\mathcal{D}_m\}_{m=1}^M$ to predict GaLR, and then employ the pretrained decoder to recover the embodiment-specific joint angles $\hat{\Theta}^m$.

IV. EXPERIMENTS

In studying cross-embodiment manipulation, we typically have two expectations. First, we hope that after co-training across different embodiments, **One embodiment can generalize to workspace regions represented in the data of other co-training embodiments.** Second, when introducing a new embodiment, we aim to **leverage cross-embodiment data to enable few-shot learning**. To this end, we design comprehensive experiments in this section to validate these two capabilities of OPFA.

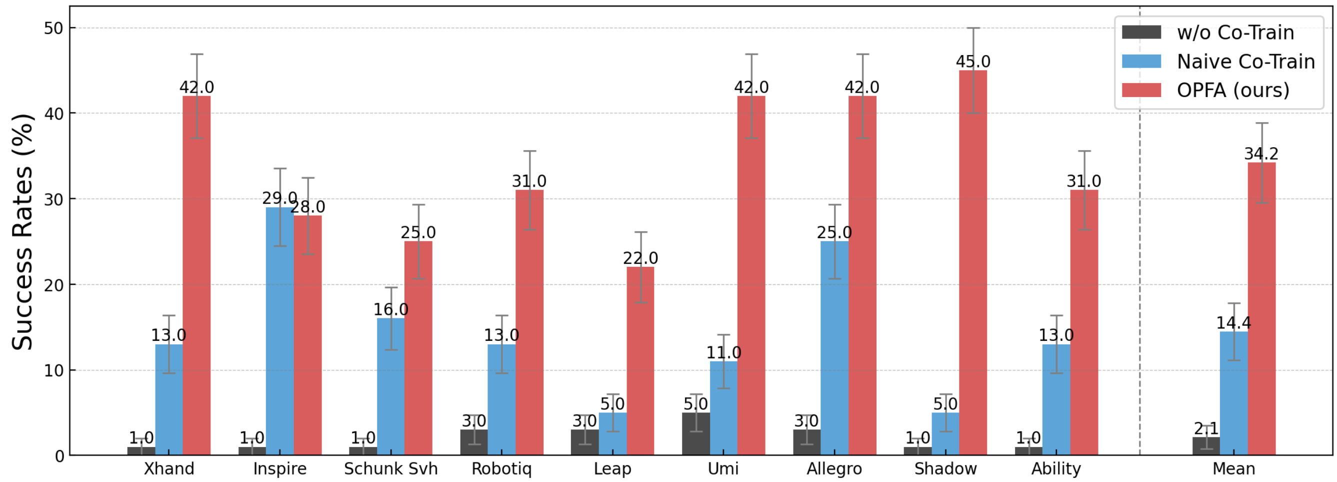

Fig. 3: **Spatial generalization evaluation on the spray-picking task.** Data for each embodiment are collected in distinct regions, and we evaluate the policy for each embodiment to generalize to regions covered by data from the others.

TABLE I: **Zero-shot skill transfer experiments.** In the table, *Position Generalization* refers to different data collection regions for different embodiments, while *Object Generalization* refers to different manipulation objects.

Task	Kettle	Button	Sanitizer	Bucket	Banana	**Pick Spray&Can**
	Spatial Generalization					Object Generalization
Inspire Hand Success Rate (%)						
w/o Co-training	30.0	26.0	5.0	3.0	10.0	1.0
Naive Co-training	57.0	39.0	71.0	50.0	83.0	57.0
OPFA (ours)	**83.0**	**60.0**	**82.0**	**75.0**	**98.0**	**83.0**
Xhand Success Rate (%)						
w/o Co-training	3.0	11.0	4.0	1.0	5.0	41.0
Naive Co-training	5.0	38.0	**61.0**	33.0	30.0	53.0
OPFA (ours)	**7.0**	**75.0**	51.0	**94.0**	**67.0**	**71.0**

A. Implementation Details

Embodiments. Our experiments involve a total of 11 embodiments. In real-world settings, we use XHand [21], Inspire Hand (tactile version) [20], Robotiq-2F-85, and Leap Hand [19], while in simulation, we additionally evaluate on UMI [18], Robotiq-3F, Allegro, Shadow Hand, Ability Hand, Schunk SVH Hand, and the non-tactile version of Inspire Hand. All embodiments share the same encoder, decoder, and latent action space.

Tasks. In simulation, we evaluate OPFA on seven tasks, including kettle-pulling, button-pressing, bucket-lifting, and pick&place. In real-world settings, we also conduct experiments on seven tasks.

Baselines. To evaluate OPFA's cross-embodiment capabilities, we compare against two baselines. The first trains only on data from the tested embodiment, denoted as *w/o Co-Train*. The second naively assigns a separate decoder to each embodiment (most current methods use) and performs cross-embodiment co-training, denoted as *Naive Co-Train*.

B. Generalization across Different Embodiments

In this section, our goal is to verify that after co-training, each embodiment can generalize within the region covered by data from the other embodiments, or can generalize to similar skills. To this end, we conduct two types of evaluations: spatial generalization and object generalization.

In the **spatial generalization** setup, we collect 72 training trajectories for each embodiment through teleoperation, and the training data of different embodiments are drawn from distinct spatial regions. We then co-train on data from multiple embodiments and evaluate each embodiment within the other embodiments' data distribution regions. Table I presents the results of Inspire Hand [20] and XHand [21] under these experimental setups.

Since the test regions are entirely unseen for each embodiment, the w/o co-train baseline almost completely fails, exhibiting poor positional generalization across five tasks with both embodiments. In contrast, the naive cross-embodiment co-train baseline partially alleviates this issue, as training on multiple embodiments provides broader wrist-level shared knowledge. However, due to the embodiment gap, transferring the fine-grained end-effector spatial information is extremely challenging, and using separate embodiment-specific decoders exactly discards this information. By comparison, OPFA co-trains multiple embodiments within a shared, geometry-aware latent action space, enabling the joint exploitation of both wrist-level and end-effector-level information. Consequently, OPFA achieves strong generalization even on entirely unseen test data, consistently demonstrating stable cross-embodiment spatial generalization across tasks and attaining success rates above 90% on tasks such as bucket-lifting and banana-picking.

Moreover, we evaluate OPFA and the baselines on their ability for cross-object skill transfer, which is more challenging. Specifically, we collect data of Inspire Hand grasping a can and XHand grasping a spray, co-train the two datasets, and then test OPFA on cross-object generalization. As shown in Table I, OPFA significantly outperforms the baselines, achieving a 26% improvement over naive co-train on Inspire Hand and an 18% improvement on XHand.

More Embodiments. To evaluate the generality of GaLR, we conduct cross-embodiment co-training experiments on nine end-effectors for a spray-picking task. The test region is divided into nine subregions, with each embodiment's training data drawn from only a single subregion. As shown

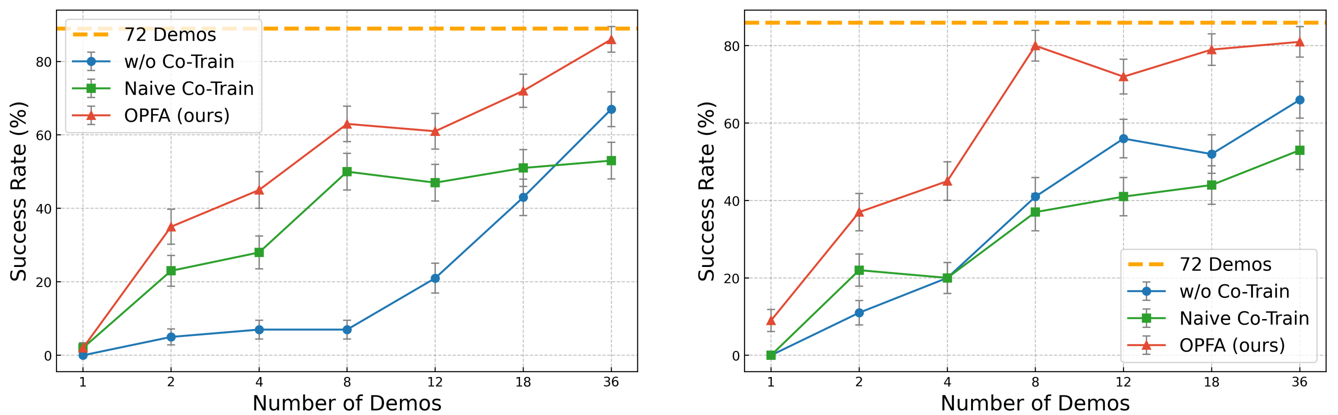

Fig. 4: **Few-shot learning curves with different demo numbers on the banana-picking task of (Left) Inspire Hand and (Right) XHand**. Each embodiment is co-trained with 72 demonstrations from the other.

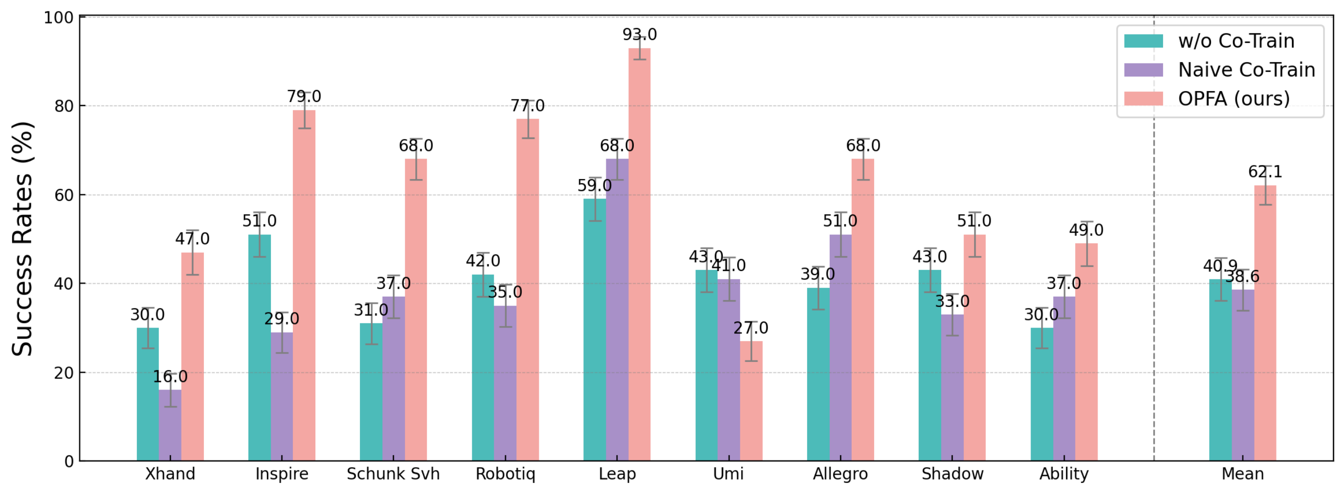

Fig. 5: **Few-shot learning performance across nine different embodiments on the spray-picking task.** We collect eight trajectories for each of the nine end-effectors and co-train on the full dataset to evaluate the few-shot learning capability for each embodiment. OPFA yields a 20%+ average performance gain over the baselines.

in Figure 3, under this challenging setup, the w/o co-train baseline completely fails, whereas OPFA significantly improves success rates for all embodiments, achieving an average improvement of 19.8% over the naive co-train method.

C. Few-Shot Learning Ability

Another key capability of OPFA is few-shot learning—that is, the ability to achieve performance close to single-source, large-data training when a new end-effector is introduced with only a small amount of data. To evaluate this, we conduct experiments on the banana-picking task, collecting 72 trajectories each for the Inspire Hand and XHand. Subsets of these trajectories are then sampled for few-shot learning tests. For the Inspire Hand, subsets of 1, 2, 4, 8, 12, 18, and 36 trajectories are co-trained with all 72 trajectories of XHand, and vice versa for XHand. The resulting few-shot learning curves are shown in Figure 4. OPFA's performance improves rapidly with the number of sampled trajectories: with only 8 trajectories, the Inspire Hand already achieves a high success rate, while XHand surpasses 80%, approaching the performance of fully trained models and far exceeding baseline methods. In contrast, naive co-training suffers from training inhibition as the sample size increases, sometimes performing worse than the w/o co-train baseline. These results demonstrate OPFA's strong few-shot learning capability.

More Embodiments. As shown in Figure 4, OPFA substantially enhances cross-embodiment few-shot learning, achieving competitive success rates with as few as eight newly collected trajectories. To further validate OPFA's few-shot capability across a wide range of end-effectors, we expand the number of embodiments to nine. In the spray-picking task, only eight trajectories are collected for each embodiment, and the combined set of 72 trajectories is used for co-training. The results, presented in Figure 5, reveal that when the number of embodiments is large, the naive co-train method may suffer from conflicts due to the significant geometric gaps between different grippers and dexterous hands, in some cases even underperforming the w/o co-train baseline. In contrast, OPFA leverages geometry-aware co-training to effectively integrate information across embodiments, leading to substantial accuracy improvements—for example, achieving a 93% success rate on Leap Hand and an average success rate of 62.1%, which represents a gain of over 20% compared to the naive co-train method.

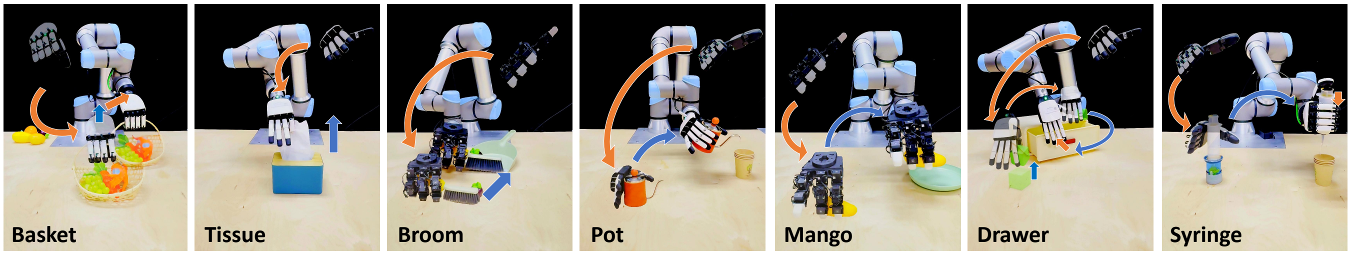

Fig. 6: Tasks for real world evaluation. *Basket* and *Mango* are one-stage pick-and-place tasks. *Tissue* and *Pot* are contact-intensive tasks, involving deformable object handling or precise manipulation. *Broom* and *Drawer* are long-horizon sequential tasks involving tool-use and multi-step coordination. *Syringe* is a fine-motor control task demanding precise force application.

TABLE II: Real-world experiments success rates. Best and second-best methods are highlighted with different colors.

Task	Basket	Tissue	Broom	Pot	Mango	Drawer	Syringe
Inspire Hand Success Rate (%)							
w/o Co-training	90	**70**	70	80	60	60	-
Naive Co-training	90	**70**	80	70	80	70	-
OPFA (ours)	**100**	60	**100**	**100**	**100**	**80**	-
Leap Hand Success Rate (%)							
w/o Co-training	90	70	80	40	60	70	60
Naive Co-training	80	50	**90**	30	10	60	70
OPFA (ours)	**100**	**90**	**90**	**80**	**90**	**90**	**100**
Robotiq-2F Success Rate (%)							
w/o Co-training	30	90	-	50	70	60	-
Naive Co-training	**90**	70	-	30	70	80	-
OPFA (ours)	**90**	**100**	-	**60**	**80**	**100**	-
Xhand Success Rate (%)							
w/o Co-training	80	80	70	70	60	50	60
Naive Co-training	80	70	80	60	60	60	70
OPFA (ours)	**90**	**100**	**100**	**90**	**100**	**90**	**100**

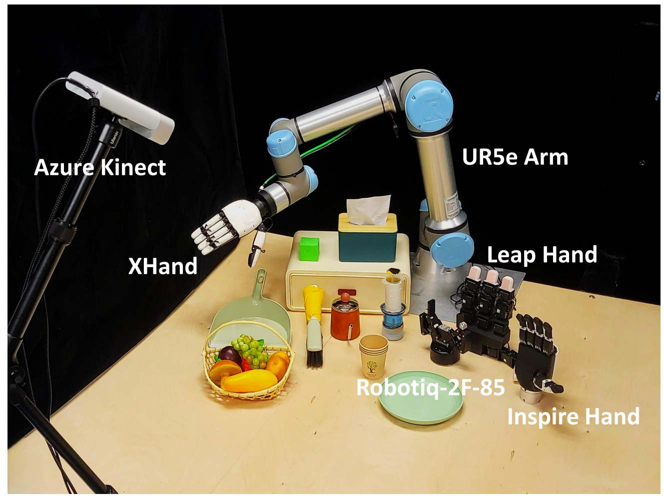

Fig. 7: Real world experiment setup.

D. Real World Experiments

Experimental Setup. Our real-world setup consists of a UR5e arm and a Microsoft Azure Kinect depth camera (Figure 7), equipped with four different end-effectors: XHand, Inspire Hand, Leap Hand, and the Robotiq-2F gripper. We design seven manipulation tasks involving diverse objects and objectives: **Basket**, pick and place a basket; **Tissue**, pull a tissue from a dispenser; **Broom**, sweep an object into a dustpan with a broom; **Pot**, grasp a pot and pour into a cup; **Mango**, pick and place a mango onto a plate; **Drawer**, place a block into a drawer and close it; **Syringe**, position a syringe over a cup and press the plunger.

For each embodiment and task, 24 demonstrations were collected via teleoperation using a Spacemouse [41] and an exoskeleton device [42]. During evaluation, **one unified policy checkpoint** is tested across all embodiments, with 10 trials per task. Due to morphological limitations, some embodiments were excluded from certain tasks (e.g., Robotiq-2F from *Syringe*/*Pot*, Inspire Hand from *Syringe*).

Results. The quantitative results of our real-world experiments are summarized in Table II. Our method demonstrates consistently high success rates across most of the evaluated tasks and embodiments, validating its capability to learn a versatile and generalizable manipulation policy. Notably, on challenging long-horizon tasks such as *Drawer* and tasks requiring fine-grained control such as *Pot*, methods trained without co-training can only learn conceptual knowledge from single-embodiment data, resulting in large operational errors when object positions vary widely. Naive co-training improves performance on long-horizon tasks to some extent, but on the *Pot* task, substantial structural differences across end-effectors induce action conflicts, leading to worse performance than w/o co-train. In contrast, OPFA enables shared learning of both wrist-level trajectory concepts and fine-grained actions across embodiments, consistently outperforming baseline methods. By sharing geometric knowledge across different end-effectors, **OPFA** achieves a level of performance that embodiment-specific heads cannot, especially in data-efficient settings. Moreover, OPFA's skill transfer capability is task-agnostic: it can generalize across both deformable (Tissue) and rigid (Broom) objects, and across both pick&place (Mango) and dexterous (Syringe) tasks.

V. CONCLUSIONS

We introduce One-Policy-Fits-All (OPFA), a unified framework for cross-embodiment manipulation that enables end-to-end co-training across data from diverse robotic hands and grippers. OPFA first learns a Geometry-Aware Latent Representation (GaLR), which constructs a shared latent action space to capture commonalities across different embodiments. It then employs a unified latent retargeting decoder to map this latent representation into embodiment-specific actions without any per-embodiment tuning. Extensive experiments on 11 end-effectors show that OPFA substantially enhances skill transfer and data efficiency, significantly improving the performance of each embodiment while exhibiting strong few-shot learning capabilities.

Acknowledgements. This work is supported by Shanghai Artificial Intelligence Laboratory.

REFERENCES

[1] D. A. Pomerleau, “Alvinn: An autonomous land vehicle in a neural network,” *Advances in neural information processing systems*, vol. 1, 1988.

[2] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[3] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in neural information processing systems*, vol. 29, 2016.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, and A. Rodriguez, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[5] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, “Llava-med: Training a large language-and-vision assistant for biomedicine in one day,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 28 541–28 564, 2023.

[6] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, *et al.*, “Qwen technical report,” *arXiv preprint arXiv:2309.16609*, 2023.

[7] 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.

[8] 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.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, and S. Jakubczak, “π_0: A vision-language-action flow model for general robot control,” *arXiv preprint arXiv:2410.24164*, 2024.

[10] Z. Zhou, Y. Zhu, J. Wen, C. Shen, and Y. Xu, “Vision-language-action model with open-world embodied reasoning from pretrained knowledge,” *arXiv preprint arXiv:2505.21906*, 2025.

[11] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, and J. Luo, “Octo: An open-source generalist robot policy,” *arXiv preprint arXiv:2405.12213*, 2024.

[12] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng, “Dexvla: Vision-language model with plug-in diffusion expert for general robot control,” *arXiv preprint arXiv:2502.05855*, 2025.

[13] E. Bauer, E. Nava, and R. K. Katzschmann, “Latent action diffusion for cross-embodiment manipulation,” *arXiv preprint arXiv:2506.14608*, 2025.

[14] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “Kpconv: Flexible and deformable convolution for point clouds,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[15] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, “Geometric transformer for fast and robust point cloud registration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 143–11 152.

[16] 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*, 2023.

[17] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations,” *arXiv preprint arXiv:2403.03954*, 2024.

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

[19] K. Shaw, A. Agarwal, and D. Pathak, “Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning,” *arXiv preprint arXiv:2309.06440*, 2023.

[20] Inspire Robots, “Inspire hand rh56dfx series,” https://inspire-robots.store/collections/the-dexterous-hands/products/the-dexterous-hands-rh56dfx-series?variant=42735794422004, accessed: 2025-09-13.

[21] Robotera, “Xhand,” https://www.robotera.com/en/goods1/4.html, accessed: 2025-09-13.

[22] A. Depierre, E. Dellandréa, and L. Chen, “Jacquard: A large scale dataset for robotic grasp detection,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3511–3516.

[23] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, “Bridge data: Boosting generalization of robotic skills with cross-domain datasets,” *arXiv preprint arXiv:2109.13396*, 2021.

[24] C. Eppner, A. Mousavian, and D. Fox, “Acronym: A large-scale grasp dataset based on simulation,” 2020. [Online]. Available: https://arxiv.org/abs/2011.09584

[25] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu, “Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot,” 2023. [Online]. Available: https://arxiv.org/abs/2307.00595

[26] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, and et al., “Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation,” 2018. [Online]. Available: https://arxiv.org/abs/1806.10293

[27] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” 2016. [Online]. Available: https://arxiv.org/abs/1603.02199

[28] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, “On bringing robots home,” 2023. [Online]. Available: https://arxiv.org/abs/2311.16098

[29] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, and et al., “Bridgedata v2: A dataset for robot learning at scale,” 2024. [Online]. Available: https://arxiv.org/abs/2308.12952

[30] A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, and A. Tung, “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)*. IEEE, 2024, pp. 6892–6903.

[31] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2023. [Online]. Available: https://arxiv.org/abs/2212.06817

[32] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” *arXiv preprint arXiv:2307.15818*, 2023. [Online]. Available: https://arxiv.org/abs/2307.15818

[33] T. Wang, D. Bhatt, X. Wang, and N. Atanasov, “Cross-embodiment robot manipulation skill transfer using latent space alignment,” 2024. [Online]. Available: https://arxiv.org/abs/2406.01968

[34] Z. Xu, B. Qi, S. Agrawal, and S. Song, “Adagrasp: Learning an adaptive gripper-aware grasping policy,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4620–4626.

[35] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, “Unigrasp: Learning a unified model to grasp with multifingered robotic hands,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.

[36] J. Romero, D. Tzionas, and M. J. Black, “Embodied hands: Modeling and capturing hands and bodies together,” *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, Nov. 2017. [Online]. Available: http://doi.acm.org/10.1145/3130800.3130883

[37] K. Shaw, S. Bahl, and D. Pathak, “Videodex: Learning dexterity from internet videos,” 2022. [Online]. Available: https://arxiv.org/abs/2212.04498

[38] H. Yuan, B. Zhou, Y. Fu, and Z. Lu, “Cross-embodiment dexterous grasping with reinforcement learning,” *arXiv preprint arXiv:2410.02479*, 2024.

[39] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, “Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation,” *arXiv preprint arXiv:2408.11812*, 2024.

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

[41] 3DConnexion, “3dconnexion space mouse,” https://3dconnexion.com/cn/product/spacemouse-compact/, accessed: 2025-09-13.

[42] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang, “Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit,” *arXiv preprint arXiv:2502.13013*, 2025.