ImMimic: Cross-Domain Imitation from Human Videos via Mapping and Interpolation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning robot manipulation from abundant human videos offers a scalable alternative to costly robot-specific data collection. However, domain gaps across visual, morphological, and physical aspects hinder direct imitation. To effectively bridge the domain gap, we propose **ImMimic**, an embodiment-agnostic co-training framework that leverages both human videos and a small amount of teleoperated robot demonstrations. ImMimic uses Dynamic Time Warping (DTW) with either action- or visual-based mapping to map retargeted human hand poses to robot joints, followed by MixUp interpolation between paired human and robot trajectories. Our key insights are (1) retargeted human hand trajectories provide informative action labels, and (2) interpolation over the mapped data creates intermediate domains that facilitate smooth domain adaptation during co-training. Evaluations on four real-world manipulation tasks (Pick and Place, Push, Hammer, Flip) across four robotic embodiments (Robotiq, Fin Ray, Allegro, Ability) show that ImMimic improves task success rates and execution smoothness, highlighting its efficacy to bridge the domain gap for robust robot manipulation. The project website can be found at https://sites.google.com/view/immimic.

Keywords: Learning from Human, Imitation learning, Dexterous Manipulation

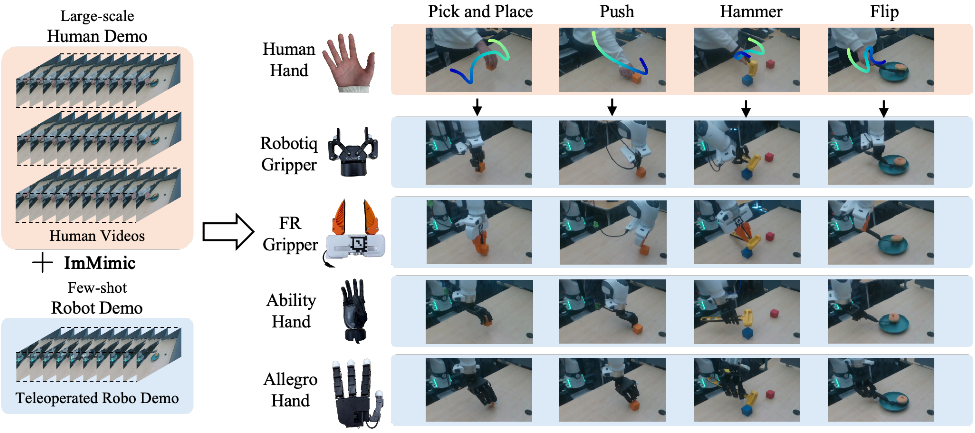


Figure 1: **ImMimic** enables embodiment-agnostic co-training between human and robot demonstrations. It leverages large-scale human videos and a small amount of teleoperated robot data, using a MixUp interpolation to enable smooth domain transfer. We validate ImMimic on four diverse manipulation tasks across four robotic embodiments.

1 Introduction

Teaching robots to perform diverse manipulation tasks in real-world environments remains a significant challenge because collecting robot-specific demonstration data is expensive. As an alternative, human videos have emerged as a promising resource, offering abundant examples of people engaging in everyday manipulation activities [1, 2]. Leveraging these human videos for robot training provides a scalable and cost-effective approach to enhance robotic skills without extensive robot demonstration collection or simulation [3–5]. However, learning robot skills from human demonstration videos still faces an inherent limitation: a substantial domain gap arising from stark differences in visual appearance, embodiment structure, physical constraints, and other factors.

In general, the challenge of enabling robots to learn human demonstrations can be formulated as a domain adaptation problem: the robot (representing the target domain) aims to emulate the behaviour of the human demonstrator (representing the source domain). A relevant application of this concept can be seen in several recent, well-established vision-based teleoperation systems [6–8] where the human demonstrator often first practices with the teleoperation setup before being able to collect high-quality robot data. This process indeed reflects an instance of *inverse* adaptation, where the human demonstrator adapts to the robot system rather than the way around. However, such inverse adaptation is absent in human demonstration videos, as human demonstrators do not consider the robot's subsequent operation. Therefore, to enable effective adaptation when learning robot skills from such human data, recent works often preprocess the input data of both domains—for example, masking out embodiments in images [4, 9] to mitigate visual differences, or restricting the action space to only 3D translations to address the embodiment-specific action gap [3, 10]. Additionally, another line of works encourages the adaptation of the latent spaces of visual inputs from both domains using unsupervised learning objectives [11–14], but often overlooks human actions, i.e., the hand trajectories, instead learning the action decoder solely from robot demonstrations.

To develop a more generalizable adaptation method across diverse robot embodiments and manipulation tasks, we introduce **ImMimic (Interpolation-via-Mapping Mimic)**, an embodiment-agnostic co-training framework that learns jointly from human demonstration videos and robot teleoperations. Our key insights are: (1) beyond the visual contexts, the retargeted human hand trajectories can serve as action labels for human demonstrations, (2) creating intermediate domains via interpolation leads to robust adaptation, and (3) establishing an effective mapping between human and robot data for interpolation is essential for co-training. Specifically, we begin by retargeting human hand poses into the robot's action space. We then perform sequence-level mapping via Dynamic Time Warping (DTW), using either visual features or action distance to pair each human timestep with its best-matching robot timestep. Finally, inspired by MixUp-based adaptation [15, 16], we interpolate both condition and predicted actions along these DTW mapping, enabling adaptation through intermediate (human–robot) domains.

To demonstrate the benefits of the ImMimic, we conduct comprehensive experiments across four different types of embodiments: Robotiq Gripper, FR Gripper [17–19], Allegro Hand, and Ability Hand. These embodiments are evaluated on four manipulation tasks: Pick and Place, Push, Hammer, and Flip. We show that ImMimic achieves higher success rates and smoother motions across all embodiments and tasks compared to baseline methods. We observe that action-based mapping provides greater improvements than visual-based mapping, suggesting that the rich action information of the human hand trajectories is equally, if not more, beneficial for co-training. Furthermore, we find that performance improves when the average action distance between human and robot is smaller, and interestingly, due to the factors such as hand-arm mounting conditions and arm kinematics, solely using a human-mimetic end-effector does not necessarily result in smaller action distances. Finally, we analyze failure cases in terms of hardware structure and algorithmic factors.

2 Related Work

Learning from Human Videos. Human videos offer an efficient and scalable source of supervision for robot learning. Retrieval-based approaches [11, 12, 14, 20, 21] search large video corpora for sequences that resemble the desired behavior for augmented learning, while typically relying on the robot's own data for action decoding. To better utilize human videos, two-stage methods [3, 5, 22–25] first learn high-level policies on human data and then adapt to robot demonstrations, but limit

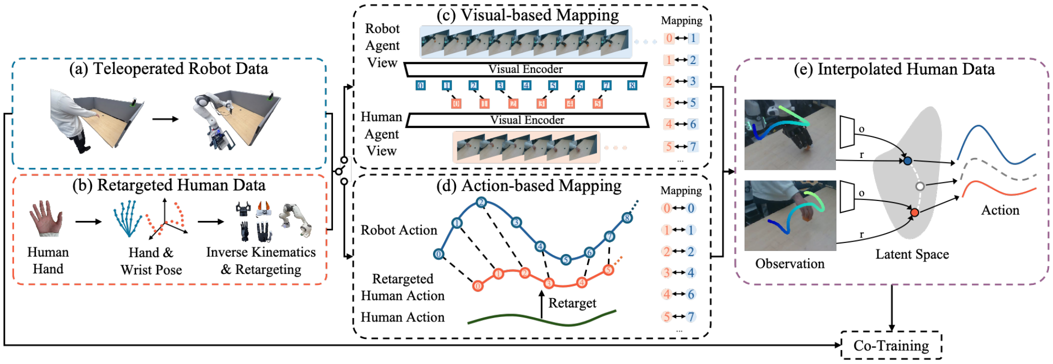


Figure 2: Overview of how we collect, map, and interpolate human and robot data. (a) Robot demonstrations are collected via visual teleoperation. (b) Human actions are retargeted from videos. (c, d) Visual or action based DTW maps retargeted human and robot trajectories. (e) MixUp: Mapped human-robot pairs are interpolated in both the latent space and action space to generate interpolated human data. Finally, we co-train the interpolated human data together with the robot data. See Fig. 3 for the co-training pipeline.

low-level action learning. To address this, co-training [4] jointly optimizes on human and robot data, yet typically relies on heavy visual preprocessing or simplified action spaces, leaving the core domain shift largely unaddressed. Our work adopts a co-training pipeline, and treating human videos as domain-adaptive supervision to smoothly address the domain shifts.

Dexterous Retargeting. Dexterous retargeting translates human hand poses into robot joint poses, ensuring that human trajectories are mapped into the same action space used for robot execution. Most recent works adopt this technique to generate robot action commands for teleoperation by observing a human demonstrator moving their own hand [6–8, 26, 27]. However, continuous guidance from the human demonstrator is crucial for the robot's success during teleoperation [28]. Other works [5, 29–36] leverage retargeted robot trajectories to learn robot policies through bespoke refinements tailored to dexterous grasping [29, 30], reinforcement learning [32–35], fine-tuning with additional robot data [5, 31], or human-in-the-loop corrections [36]. This additional process underscores the domain gap between retargeted trajectories and actual robot execution, which arises from various factors such as visual, kinematic, and physical differences. Our work handles such gap through a novel embodiment-agnostic co-training framework that smoothly adapts human demonstrations to the robot domain.

Domain Adaptation. To bridge domain gap, classic methods include adversarial feature adaptation [37] and cycle-consistent data translation [38]. Recent work inserts intermediate domains via domain flow [39] with MixUp [15, 40, 41] to build the source-to-target path. In robotic imitation learning, methods learn domain-invariant representations—such as viewpoint-agnostic and visual-invariant encoders—to handle sim-to-real and human-to-robot perception gaps [42, 43]. Meta-learned and latent-policy adaptation approaches enable rapid embodiment transfer and observation-to-action alignment [44, 45]. Structural adaptation via optimal transport or point-cloud matching, combined with modality-invariant representations and MixUp-augmented offline RL [46], helps bridge domain gap [47–50]. We extend this line by applying MixUp to visual features and actions in latent space, yielding a continuous bridge from the human domain to the robot domain.

3 Embodiment-Agnostic Co-Training Framework

We aim to learn robotic manipulation policies from large-scale, easily collected human videos, with only a small number of teleoperated robot demonstrations. For each task, the model has access to a large corpus of human demonstrations $\{\mathbf{I}_t^{\mathrm{a,h}}\}_{t=1}^{T}$, where each frame $\mathbf{I}_t^{\mathrm{a,h}} \in \mathbb{R}^{H\times W\times 3}$ is an agent-view RGB image. In parallel, it also receives a smaller set of robot demonstrations, each containing agent-view video $\{\mathbf{I}_t^{\mathrm{a,r}}\}_{t=1}^{T}$, wrist-view video $\{\mathbf{I}_t^{\mathrm{w,r}}\}_{t=1}^{T}$, and proprioception $\{\mathbf{r}_t\}_{t=1}^{T}$, where $\mathbf{r}_t \in \mathbb{R}^D$ includes end-effector pose and finger joint positions.

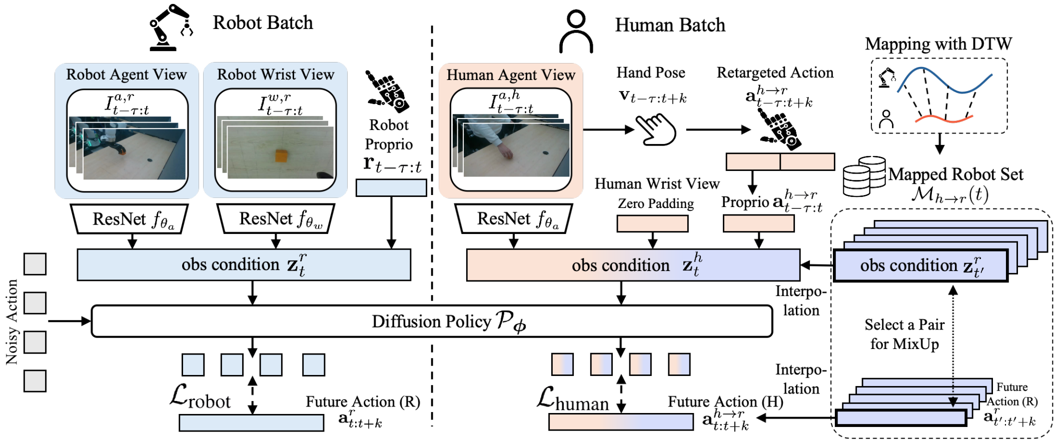


Figure 3: Overview of our embodiment-agnostic co-training framework **ImMimic**. For **robot demonstration**, we train the policy using agent- and wrist-view images encoded with ResNet f_{θ_a}, f_{θ_w}, along with proprioception $\mathbf{r}_t$. All are combined into the observation condition $\mathbf{z}^r_t$ to predict future actions. For **human demonstration**, we train the same diffusion policy $\mathcal{P}_\phi$ using human videos. A hand pose retargeting module generates retargeted robot actions $\mathbf{a}^{h\rightarrow r}_t$, which serve as both the future action and proprioception for training. Mapping with DTW, we apply MixUp (Fig. 2(e)) for human data with paired robot data. The interpolation enables human data to smoothly adapt to the robot data. The model is optimized upon the sum of reconstruction losses $\mathcal{L}_{\text{human}}$ and $\mathcal{L}_{\text{robot}}$.

This setting presents a domain adaptation challenge in the context of human-to-robot imitation learning. Specifically, human (source domain) and robot (target domain) data differ in both visual and action: (1) a visual covariate shift between human and robot observations due to embodiment appearance differences, and (2) an action gap arising from differences in embodiment structure and physical constraints, which can lead to variations in how the same task is performed. Our goal is to bridge these gaps to better adapt human demonstrations to robot execution, enabling the policy to effectively leverage both large-scale human videos and few-shot robot demonstrations.

To achieve this, we first retarget estimated human hand trajectories from videos to the robot trajectories (Sec. 3.1). We then jointly train the policy on both human and robot demonstrations (Sec. 3.2). During co-training, to achieve a smooth domain adaptation from human to robot, we pair human-robot samples using DTW, and further perform MixUp with interpolation over mapped pairs (Sec. 3.3). An overview framework is shown in Fig. 3.

3.1 Hand Pose Retargeting System

To fully leverage human videos, we extract both visual context and human hand trajectories, and then retarget hand trajectories to robot embodiments following recent advanced methods [5, 6].

Hand and Wrist Pose Estimation. We use MediaPipe [51] to localize and crop the human hand in each frame. Each patch is fed into FrankMocap [52], whose SMPL-X regressor produces precise 3D positions for 21 hand joints in the local wrist frame. By projecting these joints into the depth map and solving a Perspective-n-Point problem, we recover the wrist 6D pose in camera frame.

Retargeting. Following AnyTeleop [6], we map human keypoints $\mathbf{p}^i_t$ to robot joint angles $\mathbf{q}_t$ via

$$\min_{\mathbf{q}_t} \sum_{i=1}^{N} \left\| \alpha\, \mathbf{p}^i_t - f_i(\mathbf{q}_t) \right\|^2 + \beta \left\| \mathbf{q}_t - \mathbf{q}_{t-1} \right\|^2, \quad \mathbf{q}_l \le \mathbf{q}_t \le \mathbf{q}_u, \tag{1}$$

where f_i is the robot's forward-kinematics, and α, β balance scale and temporal smoothness.

3.2 Co-Training

While prior work often treats human videos as auxiliary pretraining data [3, 5], recent studies such as EgoMimic [4] demonstrate the benefits of co-training on both human and robot data. Motivated by this, we adopt a co-training strategy that treats human and robot demonstrations equally, allowing the policy to learn from both domains throughout a single training. Regarding the policy backbone, our framework builds on the Diffusion Policy [53].

Robot Prediction Loss. At each timestep t, where images and proprioception are denoted as $\{(\mathbf{I}_t^{\text{a,r}}, \mathbf{I}_t^{\text{w,r}}, \mathbf{r}_t)\}$, we incorporate temporal context using a history of length τ. Thus, the robot condition at each timestep is: $\mathbf{z}_t^r = \left[\mathbf{z}_{t-\tau:t}^{\text{a,r}} \,\|\, \mathbf{z}_{t-\tau:t}^{\text{w,r}} \,\|\, \mathbf{r}_{t-\tau:t}\right] \in \mathbb{R}^{(d_a+d_w+d_a)\times\tau}$, where $\mathbf{z}_t^{\text{a,r}} = f_{\theta_a}(\mathbf{I}_t^{\text{a,r}})$ and $\mathbf{z}_t^{\text{w,r}} = f_{\theta_w}(\mathbf{I}_t^{\text{w,r}})$ are feature embeddings extracted by separate ResNet18 [54] encoders f_{θ_a} and f_{θ_w} respectively.

We denote the future action sequence as: $\mathbf{a} = \left(\mathbf{a}_{t+1}^{h\to r}, \ldots, \mathbf{a}_{t+k}^{h\to r}\right)$, where k is the prediction horizon. A diffusion policy $\mathcal{P}_{\boldsymbol{\phi}}$ reconstructs $\mathbf{a}$ from a noisy action $\tilde{\mathbf{a}}$ using denoising steps conditioned on $\mathbf{z}_t$. The training objective minimizes an ℓ_2 loss: $\mathcal{L}_{\text{robot}}(\boldsymbol{\phi}) = \sum_{i=1}^{k} \left\| \mathbf{a}_{t+i}^r - \hat{\mathbf{a}}_{t+i}^r \right\|_2^2$, where $\hat{\mathbf{a}}_{t:t+k}^r = \mathcal{P}_{\boldsymbol{\phi}}\left(\tilde{\mathbf{a}}_{t:t+k}^r \,\middle|\, \mathbf{z}_t^r\right)$.

Human Prediction Loss. For a human video $\{\mathbf{I}_t^{\text{a,h}}\}_{t=1}^{T_{\text{vid}}}$, at each timestep t, the condition input includes both image features and retargeted actions: $\mathbf{z}_t^h = \left[\mathbf{z}_{t-\tau:t}^{\text{a,h}} \,\middle\|\, \mathbf{0}_{t-\tau:t} \,\middle\|\, \mathbf{a}_{t-\tau:t}^{h\to r}\right] \in \mathbb{R}^{(d_a+d_w+d_a)\times\tau}$, where each $\mathbf{z}_t^{\text{a,h}} = f_{\theta_h}(\mathbf{I}_t^{\text{a,h}}) \in \mathbb{R}^{d_a}$ is extracted using the same ResNet encoder f_{θ_a}, and $\mathbf{a}_t^{h\to r}$ is the retargeted action (from Sec. 3.1). Similarly, we compute the ℓ_2 loss as $\mathcal{L}_{\text{human}}(\boldsymbol{\phi}) = \sum_{i=1}^{k} \left\| \mathbf{a}_{t+i}^{h\to r} - \hat{\mathbf{a}}_{t+i}^{h\to r} \right\|_2^2$, where $\hat{\mathbf{a}}_{t:t+k}^{h\to r} = \mathcal{P}_{\boldsymbol{\phi}}\left(\tilde{\mathbf{a}}_{t:t+k}^{h\to r} \,\middle|\, \mathbf{z}_t^h\right)$.

Co-training Loss. During co-training, each batch includes an equal proportion of robot and human data, and the total loss is the sum of both: $\mathcal{L}_{\text{total}}(\boldsymbol{\phi}) = \mathcal{L}_{\text{robot}}(\boldsymbol{\phi}) + \mathcal{L}_{\text{human}}(\boldsymbol{\phi})$.

3.3 Mapping-guided MixUp

To create a continuum of intermediate domains in latent space such that the source and target domain on a smooth manifold [16], we propose a mapping-guided MixUp method.

Mapping. To construct the affinity between human and robot demonstrations, we compute a mapping $\mathcal{M}_{h\to r}$ between human demonstration $\mathcal{D}^h$ and robot demonstration $\mathcal{D}^r$ using *Dynamic Time Warping (DTW)* [55], based on either visual or action distance. Mapping $\mathcal{M}_{h\to r}(t)$ denotes the set of given human timestep t mapped with robot timesteps across multiple demonstrations. This mapping assumes that mapped human and robot segments with similar visual or action patterns correspond to shared states [56]. Similar to retrieval-based methods [14], DTW ensures temporal consistency and avoids implausible supervision. We explore two mapping strategies: (1) *Action-based Mapping.* We define the action distance between a retargeted human demonstration and a robot demonstration as a weighted sum of several components: $d_{\text{act}} = \|\mathbf{t}^{h\to r} - \mathbf{t}^r\|_1 + \lambda_1 \|\mathbf{p}^{h\to r} - \mathbf{p}^r\|_1 + \lambda_2\, d_{\text{rot}}\left(\mathbf{o}^{h\to r}, \mathbf{o}^r\right)$, where $\mathbf{t}$ denotes the translation, $\mathbf{p}$ the hand pose, $\mathbf{o}$ the orientation, d_{rot} the angular distance and λ_1, λ_2 are weighting coefficients, and (2) *Visual-based Mapping.* Here, we compute the frame-wise distance using extracted visual features and temporal alignment: $d_{\text{vis}} = \|\mathbf{f}^{h\to r} - \mathbf{f}^r\|_2$, where $\mathbf{f}$ represents visual features extracted from a pretrained encoder.

MixUp-based Interpolation. After establishing the mapping, we apply MixUp [15] to interpolate between original human and robot data, creating interpolated human data. During co-training, we train jointly on both the interpolated human data and the original robot data, serving as both regularization and augmentation.

During training, we apply MixUp on both the condition inputs and the predicted actions. At each training iteration, for each human timestep t, we randomly sample a robot timestep $t' \in \mathcal{M}_{h\to r}(t)$ and construct the mixed condition input and predicted action as:

$$\mathbf{z}_t^{\text{mix}} = \alpha \cdot \mathbf{z}_t^h + (1-\alpha)\cdot \mathbf{z}_{t'}^r, \quad \mathbf{a}_{t:t+k}^{\text{mix}} = \alpha \cdot \mathbf{a}_{t:t+k}^{h\to r} + (1-\alpha)\cdot \mathbf{a}_{t':t'+k}^r \tag{2}$$

where $\mathbf{z}_t^h$ and $\mathbf{a}_{t:t+k}^{h\to r}$ are the condition input and retargeted action of the raw human data, and $\mathbf{z}_{t'}^r$, $\mathbf{a}_{t':t'+k}^r$ come from the mapped robot demonstration. Inspired by DLOW [16], we adopt a progressive interpolation strategy that gradually decreases the coefficient α during training, enabling smoother domain adaptation.

Setting	Pick and Place				Push			
	Robotiq	FR	Allegro	Ability	Robotiq	FR	Allegro	Ability
Robot Only	0.40	1.00	0.00	0.80	0.00	0.60	1.00	1.00
Co-Training	0.40	1.00	**1.00**	0.80	0.20	0.60	1.00	1.00
ImMimic-A	**1.00**	1.00	**1.00**	**1.00**	**0.40**	**0.70**	1.00	1.00

Setting	Hammer				Flip			
	Robotiq	FR	Allegro	Ability	Robotiq	FR	Allegro	Ability
Robot Only	0.20	0.90	0.00	0.00	0.60	0.60	0.00	0.60
Co-Training	0.40	0.80	0.00	0.00	0.60	0.80	0.00	0.90
ImMimic-A	**0.50**	**1.00**	**0.20**	0.00	**1.00**	**0.80**	**0.20**	**1.00**

Table 1: Success rates of Robot-Only, Co-Training, and ImMimic-A across four embodiments and four tasks. Policies are trained using 5 robot demonstrations and 100 human demonstrations.

3.4 Inference

At test time, actions are predicted at a fixed inference frequency in the timestep of k. At each inference step t, an upsample rate γ, which is calculated from the duration of teleoperation and consistent with the rate used in training, is applied to both observations and predicted actions (details in the Supp. A). The condition $\mathbf{z}_t^\tau$ is constructed using an observation history of length τ. A future action sequence $\mathbf{a} = (\hat{\mathbf{a}}_{t+1}, \ldots, \hat{\mathbf{a}}_{t+k})$ is predicted from random noise. For stability, temporal ensembling is applied with a decaying weight factor to average overlapping predictions across timesteps.

4 Experiment Setups

Hardware setup. We conduct experiments using a Franka Emika Panda robot arm equipped with four types of end-effectors (see Fig. 1): (1) Robotiq 2F-85 Fripper (2-finger), (2) Fin Ray Fripper (2-finger), (3) Allegro Hand (4-finger), and (4) Ability Hand (5-finger). These devices provide a range of dexterity and serve to evaluate embodiment transfer under different hardware configurations.

Tasks. We introduce two categories of manipulation tasks, desinged to target increasing levels of control difficulty and embodiment demands: *(1) Basic Object Manipulation.* These tasks assess coarse end-effector control and general spatial positioning: **Pick and Place:** The robot must pick up a cube from a random initial position and place it precisely at a designated goal location. **Push:** The robot must push a cube across a tabletop surface into a specified goal region. *(2) Tool-based Manipulation.* These tasks evaluate the robot's ability to manipulate external tools as a proxy for object interaction: **Hammer:** The robot picks up a hammer and strikes a target point. **Flip:** The robot uses a spatula to flip a bagel off the surface.

Baselines. We compare against the following baselines in our experiments: Robot-only (Training diffusion policy using only robot data), Two-stage Fine-Tuning (Pretraining on human videos, followed by fine-tuning with robot data), Vanilla Co-Training (Simultanous training on both human and robot data), Random Mapping (Randomly pairing human and robot data for MixUp), Visual Mapping (ImMimic-V, using DTW with visual feature for mapping), and Action Mapping (ImMimic-A, using DTW with action for mapping).

Metrics. We evaluate performance using three key metrics: *(1). Success Rate (SR).* The proportion of 10 rollouts that successfully complete the task, scored in a binary manner (success or failure). *(2). Trajectory Smoothness (SPARC).* Trajectory smoothness is quantified using the Spectral Arc Length (SPARC) [57], which measures the smoothness in the frequency domain. *(3). Action Distance (AD).* The average distance of translation and orientation after DTW for trajectory similarity.

5 Core Results

Human videos enhance the robustness and smoothness of learned policies. Leveraging human videos substantially improves policy success rates as shown in Tab. 1. As shown in Tab. 2, policies

Setting	Robotiq		Ability	
	Pick and Place	**Flip**	**Pick and Place**	**Flip**
Robot Only	0.40	0.60	0.80	0.60
Fine-Tuning	0.80	0.70	0.50	0.40
Co-Training	0.40	0.80	0.80	0.90
Random Mapping	0.40	0.50	0.80	0.50
ImMimic-V	**1.00**	0.50	0.90	0.70
ImMimic-A	**1.00**	**1.00**	**1.00**	**1.00**

Table 2: Comparison of success rate across two embodiments (Robotiq, Ability) and two tasks (Pick and Place, Flip), with 5 robot demos and 100 human demos.

Embodiment	Rollout (Robot Only)	Rollout (Co-Training)	Rollout (ImMimic-A)
Robotiq	-12.7694	-9.6533	**-9.4424**
FR	-24.4935	**-14.3644**	-15.6430
Ability	-13.9228	-10.9241	**-10.8409**
Allegro	N/A	-17.1312	**-13.8940**

Table 3: Spectral Arc Length (SPARC) smoothness scores (↑) on Pick and Place. A higher score indicates a smoother trajectory. We evaluate average scores on 5 successful rollouts over 3 methods.

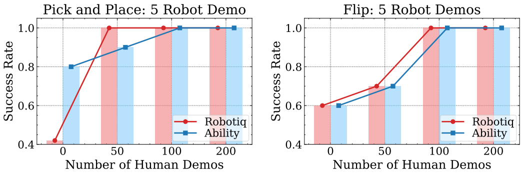


Figure 4: Sample efficiency of ImMimic-A with varying numbers of human demonstrations.

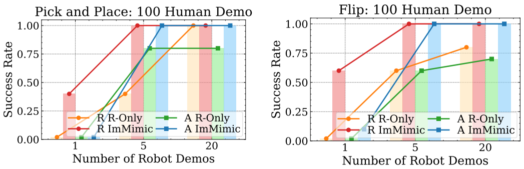


Figure 5: Sample efficiency of ImMimic-A and Robot-Only with varying numbers of robot demonstrations.

trained with ImMimic-A consistently achieve higher success rates across all tasks and embodiments compared to robot-only, two-stage fine-tuning, and co-training baselines. These results indicate that learning from human videos using our method improves the robustness of robot rollouts, as the interpolated human data effectively serves as data augmentation for the limited robot data. For example, ImMimic-A is more robust to variations in object positions (Fig. 7(c)). Furthermore, ImMimic improves action smoothness. In Tab. 3, we show that it achieves higher SPARC scores compared to robot-only policies, indicating smoother trajectories. It also outperforms vanilla co-training on three out of four embodiments. These results together suggest that our method effectively enhances the robot policy by leveraging prior knowledge from human demonstrations.

Interpolation with Action-based Mapping leads to better performance. We compare action-based and visual-based mapping to evaluate their effectiveness in bridging human-robot domain gap. As shown in Tab. 2, action-based mapping (ImMimic-A) consistently outperforms visual-based mapping (ImMimic-V) and random mapping. This performance gain is attributed to the fact that retargeted human actions, aligned via kinematic constraints, are structurally more similar to robot actions than visual features are to robot observations. In a separate long-horizon video retrieval task (details in Supp. C), we extend DTW to retrieve robot-relevant subsequences from unsegmented human videos. The results in Fig. 6 show that action-based mapping can be more accurate and robust with visual and action disturbance. Fig. 7(e) shows ImMimic-V failing due to poor mapping, causing the robot to loop in place. Especially in task with subtle action transitions, weak visual mapping degrades performance, highlighting that mapping quality is critical, and training with action-based mapping leads to a more reliable robot policy.

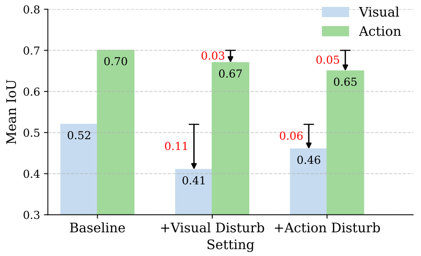


Figure 6: Comparison of Mean IoU across different disturbance settings.

ImMimic leads to consistent improvement across embodiments. ImMimic consistently enhances policy performance across different end-effectors, regardless of their morphological similarity to the human hand. As shown in Tab. 1, ImMimic-A improves task success across all embodiments compared to the Robot-Only baseline, and outperforms or matches the performance of Co-Training. This demonstrates that ImMimic-A effectively adapts to various tasks and embodiments.

However, for certain embodiment-task, success rates remain low despite using our method. For Hammer with Ability Hand (0.0 SR), Fig. 7(d) shows that the short thumb causes the index finger to

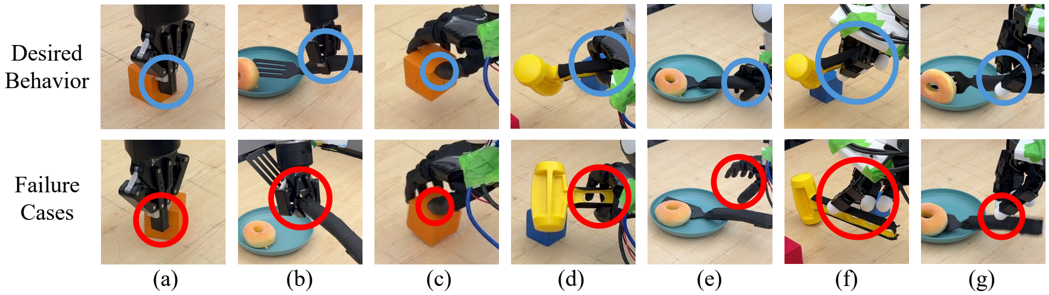


Figure 7: Desired behavior and corresponding failure cases. (a) Unstable push due to thin tip. (b) Unstable grasp from structural gap. (c) Grasp failure due to variations in object positions. (d) Poor hammer grasp from bad initial contact. (e) Motion trap due to weak visual mapping. (f) Insufficient gripping force under heavy hammer weight. (g) Infirm grasp of spatula due to large hand.

make unintended contact with the hammer, leading to misoriented grasp. For Flip with Allegro Hand ($\leq$ 0.2 SR), Fig. 7(g) shows a failure case where the hand cannot firmly grasp the spatula due to its large size. These cases show the essential effects of embodiment structure on task performance.

More human-mimetic embodiments don't necessarily lead to better transfer. Intuitively speaking, human-mimetic embodiments should exhibit smaller action distance to human demonstrations, but our results show otherwise. Average AD (Action Distance) shows that two dexterous hands demonstrate larger action distances (Allegro: 0.078, Ability: 0.075) compared to the two grippers (Robotiq: 0.066, FR: 0.065). Moreover, Tab. 1 show that policies benefit more from human videos when the action distance is smaller, regardless of its mechanical structure. This is potentially due to the fact that in addition to hand design, mounting configuration and arm kinematics also influence the action retargeting and the way robot performs the task.

These observations are likely to offer useful insights for end-effector design as well. In Fig. 7(a,b), Robotiq's thin fingertips and palm gap cause unstable contact and slipping. In Fig. 7(d), Ability's shorter thumb may contribute to misaligned grasps when position offsets are present. In Fig. 7(f,g), Allegro's larger size appears to limit its ability to lift the heavy hammer or grasp the spatula firmly. Overall, features such as longer fingertips, extended thumb reach, and higher grasping force may support more robust performance across a range of manipulation tasks.

The scale and diversity of human demonstrations enhance learning performance. Human videos exhibit greater diversity than robot data, as reflected by a higher intra-dataset Action Distance (AD) (0.012 vs. 0.005). In Fig. 4, for Pick and Place, adding 50 human videos raises success rate (SR) from 0.4 to 1.0 for the Robotiq and from 0.8 to 0.9 for the Ability; both reach 1.0 by 100 videos. Conversely, in Fig. 5, with 100 human videos fixed, ImMimic-A achieves 1.0 SR with only 5 robot demonstrations, while the robot-only baseline requires 20 demos but still underperforms. These results suggest that incorporating human data can significantly improve sample efficiency when combined with a small amount of robot data.

6 Conclusion

We present ImMimic, a novel embodiment-agnostic co-training framework that unites large-scale human videos with few-shot robot demonstrations. To bridge the domain gap between human and robot data, ImMimic leverages DTW-based mapping and MixUp to interpolate between mapped human-robot pairs, creating intermediate domains that enable smooth domain adaptation during co-training. Evaluation on four tasks and four embodiments demonstrates consistent improvements in task success rate and rollout smoothness. Additionally, we find that mapping based on action similarity between retargeted human and robot actions, rather than visual context, leads to improved policy performance, suggesting that human hand trajectories offer rich supervision for robot learning. We also identify several failure cases, attributed to either hardware design or limitations in the learning method, and observe that a more human-like hand does not necessarily yield better performance.

7 Limitations

Exisiting limitation of ImMimic includes: (1) **Large domain gap leads to performance drop.** Although ImMimic outperforms baselines across embodiments in most of the tasks, its performance is still degraded under even larger domain gaps, such as significant differences in average action distances between embodiments and humans, or major visual appearance differences. Future directions may include improved representation learning to better align the features even across larger domain gaps. (2) **Inconsistent gains across embodiments potentially indicate that policy performance is influenced by the robot's structural design.** While ImMimic consistently improves success rates and smoothness across all four embodiments shown in Tab. 1 and Tab. 2, the magnitude of these gains varies. In future work, we aim to empirically investigate how embodiment design impacts policy performance when learning from human demonstrations, with the ultimate goal of developing a unified system that enables robots to more effectively acquire and adapt human skills.

Appendix

A Demonstration Collection System

The overall data collection system is illustrated in Fig. A.1. We collect both human demonstration videos and robot teleoperation data to establish a comprehensive dataset for our study. To minimize visual gap between human and robot demonstrations, we use the same RealSense D435 camera for both. Demonstrations are recorded from a fixed viewpoint that captures the entire workspace and clearly shows hand-object interactions.

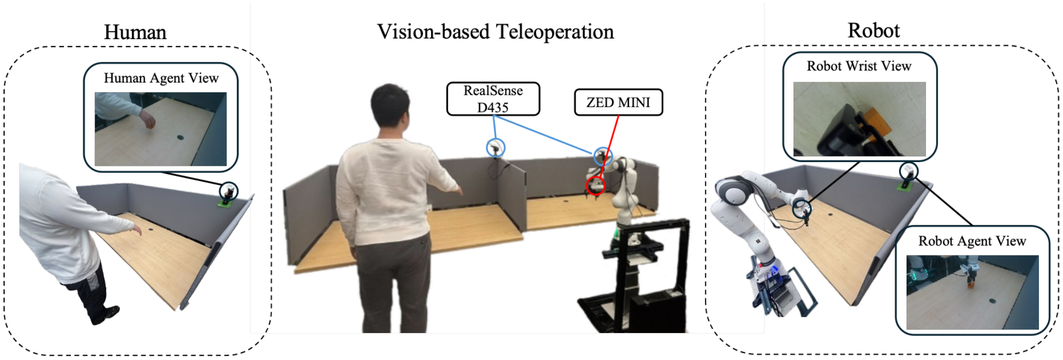


Figure A.1: Overall data collection system. (1) For human demonstrations, only the agent-view camera is used. (2) For robot demonstrations, both the agent-view and wrist-view cameras are used to enable precise control. (3) For teleoperation, a separate workspace is placed to the left of the robot, and a camera with identical intrinsics and calibration is used for vision-based control.

A.1 Data Collection Throughput

As shown in Tab. A.1, we report the teleoperation throughput for each embodiment on each task in terms of: (1) **Frequency** – the average number of successful demonstrations recorded per minute, (2) **Success Rate** – the ratio of successful demonstrations to total attempts, and (3) **Duration** – the average length of all successful demonstrations. Due to structural differences and varying task difficulty, these metrics differ across embodiments and tasks. These trends also strongly correlate with the final policy performance. For Hammer, using Allegro Hand and Ability Hand for teleoperation shows low success rates (≤ 0.3) and require longer durations due to the need for precise wrist angle adjustments during teleoperation. This aligns with the policy rollout results, where the policies learned with these embodiments also exhibit low rollout success rates (≤ 0.2). In contrast, for the same tasks, using the Robotiq Gripper and FR Gripper for teleoperation shows better performance, and the policies trained for these embodiments achieve higher performance.

A.2 Sample Rate Normalization

To enable consistent training and inference across human and robot demonstrations, we define a sample rate γ that compensates for the difference in demonstration durations. As shown in Tab. A.1, human demonstrations tend to be faster, while teleoperated robot demonstrations take longer time. To align their temporal coverage, we fix the action sequence length $k = 32$ for human demonstrations, then compute γ as the ratio of robot to human demonstration durations. Using this value, we uniformly subsample γ-spaced frames from each robot demonstration to produce a k-step sequence that spans a comparable duration.

During training, we use an observation history length $\epsilon = 2$, where the policy predicts k future actions based on ϵ past observations. For robot data, these observations are offset by γ, allowing the model to learn over a similar time horizon as in human data. This normalization helps mitigate issues caused by overly short prediction horizons in slower-paced robot trajectories.

Method	Metric	Pick and Place	Push	Hammer	Flip
Human Demo	Frequency	5.4	6.7	2.8	3.4
	Success Rate	1.00	1.00	1.00	0.98
	Duration	2.66	1.59	4.66	2.52
Vision-based Teleop (Robotiq)	Frequency	1.47	1.33	1.05	0.45
	Success Rate	0.82	0.88	0.48	0.28
	Duration	8.33	9.17	12.73	7.54
Vision-based Teleop (FR)	Frequency	1.4	1.52	0.83	0.76
	Success Rate	0.83	0.88	0.52	0.46
	Duration	12.87	17.04	16.23	11.04
Vision-based Teleop (Allegro)	Frequency	1.42	1.67	0.12	0.43
	Success Rate	0.70	0.86	0.04	0.32
	Duration	15.43	10.99	21.31	14.78
Vision-based Teleop (Ability)	Frequency	1.21	2.05	0.38	0.59
	Success Rate	0.68	0.91	0.22	0.45
	Duration	16.09	10.12	18.28	13.86

Table A.1: Frequency (number of successful demonstrations collected per minute), Success Rate (ratio of successful demonstrations) and Duration (average duration of all demonstrations) for human demonstrations and vision-based teleoperation across four tasks using four different end-effectors: Robotiq, Fin Ray, Allegro, Ability.

Method	Pick and Place	Push	Hammer	Flip
Human Demo	32	32	32	32
Robotiq	100	185	87	96
FR	155	343	112	140
Allegro	185	221	146	188
Ability	193	204	126	176

Table A.2: Sample rate γ used during training and inference. It is computed as the ratio between the durations of human and robot demonstrations and is used to subsample robot data during training and upsample predicted actions during inference.

At inference, we upsample the predicted k-step sequence using γ to recover the original robot execution speed. The model performs inference every k steps, and intermediate steps are filled via temporal ensembling of previously predicted actions with a decaying weight. This ensures smooth, continuous motion during rollout while maintaining consistency with the teleoperated control pace.

A.3 Camera Calibration

Accurate camera calibration is essential for both human and robot demonstrations. Before data collection, we calibrate the agent-view RealSense D435 camera used across our settings. For vision-based teleoperation, we use a separate RealSense D435 camera positioned over a dedicated workspace to the left of the robot for RGBD-based hand pose estimation and retargeting. This camera shares the same intrinsic parameters and calibration with the agent-view camera.

We now describe the camera calibration method used to transform retargeted human trajectories (extracted from human demonstration videos) from the camera coordinate frame to the robot base frame. Specifically, we aim to estimate the rigid transformation that maps 3D points and orientations from the camera frame to the robot base frame, denoted as:

$$^{\text{base}}\mathbf{T}_{\text{cam}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{A.1}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. In homogeneous coordinates, any point $\mathbf{p}_{\text{cam}}$ in the camera frame is transformed to the robot base frame via:

$$\begin{bmatrix} \mathbf{p}_{\text{base}} \\ 1 \end{bmatrix} = {}^{\text{base}}\mathbf{T}_{\text{cam}} \begin{bmatrix} \mathbf{p}_{\text{cam}} \\ 1 \end{bmatrix}. \tag{A.2}$$

To perform calibration, we attach an AprilTag to a known location (Fig A.2a) such that its pose relative to the robot base is known, yielding ${}^{\text{base}}\mathbf{T}_{\text{tag}}$. The camera observes the tag, yielding ${}^{\text{tag}}\mathbf{T}_{\text{cam}}$. Combining these yields:

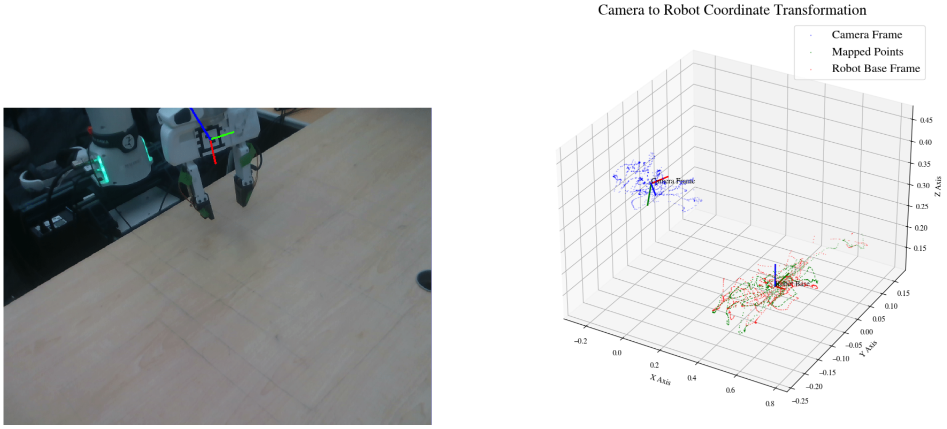
$${}^{\text{base}}\mathbf{T}_{\text{cam}} = {}^{\text{base}}\mathbf{T}_{\text{tag}} \left({}^{\text{tag}}\mathbf{T}_{\text{cam}}\right)^{-1}, \tag{A.3}$$

Multiple such measurements enable us to refine $(\mathbf{R}, \mathbf{t})$ using a best-fit procedure. Given N pairs of corresponding points $\mathbf{p}_i^{\text{cam}}$ (in the camera frame) and $\mathbf{p}_i^{\text{rob}}$ (in the robot base frame), we estimate $(\mathbf{R}, \mathbf{t})$ by minimizing:

$$\mathcal{L}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{p}_i^{\text{rob}} - (\mathbf{R}\,\mathbf{p}_i^{\text{cam}} + \mathbf{t})\|^2, \tag{A.4}$$

$$\text{s.t.} \mathbf{R}^T\mathbf{R} = \mathbf{I}. \tag{A.5}$$

We use a quaternion-based parameterization of $\mathbf{R}$ to enforce $\mathrm{SO}(3)$ constraint and solve the problem via nonlinear least squares. The overall calibration procedure is illustrated in Fig A.2b.



(a) AprilTag used for camera calibration, enabling precise estimation of its 6-DoF pose in the camera frame.

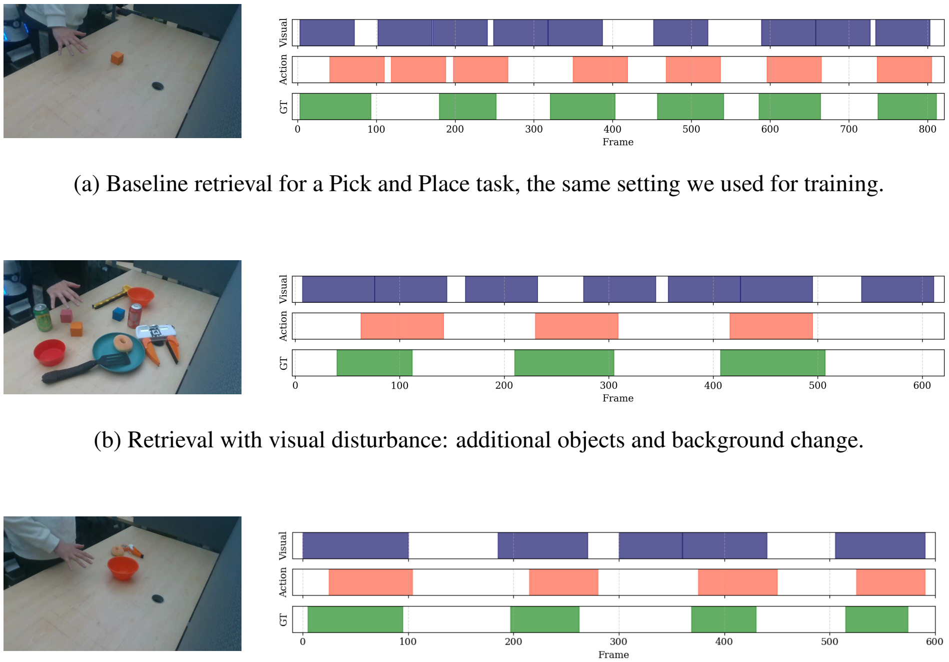
(b) Calibration from the camera coordinate frame to the robot base frame.

Figure A.2: Camera calibration process.

B Retargeting

In both human videos processing and vision-based teleoperation for robot data collection, we perform retargeting from human hand motion to robot actions. While the overall retargeting pipeline is shared across both settings, there are key differences. For human demonstration videos, we use offline retargeting based on RGB inputs and apply position retargeting, where absolute 3D joint positions are mapped to robot actions. For real-time vision-based teleoperation, we apply online retargeting that replaces wrist pose estimation with a more stable depth-based method and adopts vector retargeting [6], which aligns finger segment orientations rather than absolute positions for teleoperation. This section provides additional details on the retargeting process.

Human Pose Estimation and Wrist Localization. To estimate human hand pose, we use MediaPipe [51], a real-time pipeline that provides robust hand bounding boxes. Each cropped hand region is then passed to FrankMocap [52], which outputs shape and pose parameters for an SMPL-X model [58], resulting in accurate 3D coordinates for 21 knuckle joints in the local wrist frame.



(a) Baseline retrieval for a Pick and Place task, the same setting we used for training.

(b) Retrieval with visual disturbance: additional objects and background change.

(c) Retrieval with action disturbance: Pick and Place different objects.

Figure C.1: Comparison of visual- and action-based mapping methods under baseline, visual disturbance, and action disturbance conditions. The results indicate that visual-based mapping suffers a more noticeable performance drop under visual disturbances, while action-based mapping remains comparatively robust.

To improve spatial accuracy, particularly important for teleoperation, we replace FrankMocap's estimated wrist translation with a wrist point derived from depth data captured by an RGBD camera. For wrist orientation, we apply the **Perspective-n-Point (PnP) algorithm** [59], solving:

$$R^*, t^* = \arg\min_{R,t} \sum_i \|\mathbf{p}_i - \Pi(R\mathbf{P}_i + t)\|^2 \tag{B.1}$$

where $\mathbf{P}_i$ are the 3D keypoints in the local frame, $\mathbf{p}_i$ are their 2D projections, $R \in SO(3)$ is the orientation matrix, t is the translation vector, and Π is the camera projection function. This yields a refined 6-DoF wrist pose that is consistent with the observed depth.

Online Retargeting for Real-time Teleoperation. For real-time teleoperation, we adopt vector retargeting to ensure responsiveness and avoid kinematic singularities. Instead of matching absolute joint positions, we optimize finger orientations to follow the directions of human keypoint vectors. Given keypoint vectors $\mathbf{v}_t^i$ from MediaPipe [51], we solve for the robot joint configuration by minimizing:

$$\min_{\mathbf{q}_t} \sum_{i=1}^{N} \left\|\alpha\, \mathbf{v}_t^i - \mathbf{R}\, f_i(\mathbf{q}_t)\right\|^2 + \beta \left\|\mathbf{q}_t - \mathbf{q}_{t-1}\right\|^2, \quad \text{s.t.} \quad \mathbf{q}_l \leq \mathbf{q}_t \leq \mathbf{q}_u, \tag{B.2}$$

where $f_i(\cdot)$ maps to the corresponding robot finger vector, $\mathbf{R}$ aligns coordinate frames, and α, β control the scaling and temporal smoothness. We solve this constrained optimization problem in under 10 ms per frame. To further reduce latency and improve motion continuity, we apply a low-pass filter with a smoothing parameter of 0.2 to suppress sudden keypoint fluctuations. This enables stable control and recording at 30 Hz.

C Long Raw Human Video Retrieval

C.1 Greedy Multi-Segment Subsequence DTW (GMS-SDTW).

In our current setup, we use segmented human and robot demonstrations recorded in the same workspace while performing the same task. This controlled design minimizes the visual and action gap and simplifies the mapping process. In contrast, more practical scenario involves long, untrimmed human videos that include disturbances and task-irrelevant actions. In such cases, identifying an accurate mapping strategy becomes even more critical. To extract useful segments from these raw videos, recent retrieval-based methods attempts to match human segments with corresponding robot behaviors, most often relying on visual features [14]. We formulate a retrieval task using long human videos, enabling a comparison between visual- and action-based mapping strategies to clarify which modality yields higher accuracy. To address this, we propose **Greedy Multi-Segment Subsequence DTW (GMS-SDTW)**, an extension of our current mapping algorithm.

Overview of GMS-SDTW. Given a long human trajectory $\mathbf{H} = \{\mathbf{h}_t\}_{t=1}^{T_h}$ that contains an unknown number of action subsequences, and a single robot trajectory $\mathbf{R} = \{\mathbf{r}_s\}_{s=1}^{T_r}$, our goal is to identify *all* the human subsequences that best match the robot trajectory. We extend classical Subsequence DTW (S-DTW) by scanning through $\mathbf{H}$ using a sliding window method, greedily selecting mapped subsequences whose distance to the robot trajectory is below a predefined threshold ϵ. The sliding window length L is varied within a predefined range $L \in [L_{\min}, L_{\max}]$. The algorithm is presented in Alg. 1.

Setting	**Method**	**mIoU**	**Acc@0.5**
Baseline	Visual	0.52	66.7
	Action	0.70	71.4
+ Visual Disturb.	Visual	$0.41_{\downarrow 0.11}$	$33.3_{\downarrow 33.4}$
	Action	$0.67_{\downarrow 0.03}$	$66.7_{\downarrow 4.7}$
+ Action Disturb.	Visual	$0.46_{\downarrow 0.06}$	$40.0_{\downarrow 26.7}$
	Action	$0.65_{\downarrow 0.05}$	$75.0_{\uparrow 3.6}$

Table C.1: Comparison of mean IoU and Acc@0.5 for visual- and action-based mappings under different disturbance conditions on long raw videos.

S-DTW. The cumulative distance matrix $D(i, j)$ is initialized to support open-ended matching in the candidate sequence $\mathbf{R}$:

$$D(0,0) = d(0,0), \quad D(i,1) = \sum_{k=1}^{i} d(k,1), \quad D(1,j) = d(1,j) \qquad \text{for } i = 1, \dots, T_h \text{ and } j = 1, \dots, T_r. \tag{C.1}$$

where $d(i, j)$ is the pairwise distance between the i-th human frame and j-th robot frame.

The recursive update is follows the standard DTW formulation:

$$D(i,j) = d(i,j) + \min\{D(i-1,j-1),\, D(i-1,j),\, D(i,j-1)\}\,. \tag{C.2}$$

The best-matching endpoint is chosen as $j^\star = \arg\min_j D(T_h, j)$, and the start index is recovered via backtracking from $(i, j^\star)$.

Greedy search. Starting at frame $t = 1$, we evaluate subsequence $\mathbf{H}_{t:t+L}$ for lengths $L \in [L\text{min}, L_{\max}]$ via S-DTW, get the subsequence with the minimum distance, and store it if $d^\star < \epsilon$. Stored subsequences are recorded as segments $(t, t + L^\star, k^\star, j^\star)$ and the search resumes from $t = t + L^\star + 1$. Otherwise we increment $t \leftarrow t + 1$. The algorithm runs in $\mathcal{O}\big((L_{\max} - L_{\min})\, T_r, T_h\big)$ time, and each robot frame is only assessed within the S-DTW dynamic-programming table.

Complexity. Each S-DTW distance has a time complexity of $\mathcal{O}(T_h T_r)$. With a linear scan over T frames and at most $L_{\max} - L_{\min} + 1$ window lengths, the overall complexity is $\mathcal{O}\big((L_{\max} - L_{\min} + 1)\, T_r T_h\big)$, which is tractable in practice since $L_{\max} \ll T$.

C.2 Visual- and Action-based Long Raw Video Retrieval

As discussed in **Core Results**, action-based mapping tends to offer more robust performance than visual mapping. To further compare their performance, we propose a long raw video retrieval task [60] as an intuitive way to assess the robustness of each mapping method under varying conditions. In

Algorithm 1 Greedy Multi-Segment Subsequence DTW (GMS-SDTW)

```
Require: Human trajectory $\mathbf{H}$ of length $T_h$; robot trajectory $\mathbf{R}$ of length $T_r$;
      window bounds $L_{\min}, L_{\max}$; distance threshold $\epsilon$
Ensure: Set $\mathcal{P}$ of matched segments $(h_{\text{start}}, h_{\text{end}}, r_{\text{start}}, r_{\text{end}}, d)$
$\mathcal{P} \leftarrow \emptyset$;    $t \leftarrow 1$
while $t + L_{\min} - 1 \leq T_h$ do
    $d_{\text{best}} \leftarrow +\infty$
    for $L = L_{\min}$ to $\min(L_{\max}, T_h - t + 1)$ do
        $\mathbf{q} \leftarrow \mathbf{H}_{t:t+L-1}$
        $(d, j_{\text{start}}, j_{\text{end}}, \_) \leftarrow$ S-DTW$(\mathbf{q}, \mathbf{R})$
        if $d < d_{\text{best}}$ then
            $d_{\text{best}} \leftarrow d$
            $L^\star \leftarrow L$
            $j^\star_{\text{start}} \leftarrow j_{\text{start}}$
            $j^\star_{\text{end}} \leftarrow j_{\text{end}}$
        end if
    end for
    if $d_{\text{best}} < \epsilon$ then
        Add $(t, t + L^\star - 1, j^\star_{\text{start}}, j^\star_{\text{end}}, d_{\text{best}})$ to $\mathcal{P}$
        $t \leftarrow t + L^\star$                              ▷ Skip matched subsequence
    else
        $t \leftarrow t + 1$
    end if
end while
return $\mathcal{P}$
```

addition to segmented human demos with well-defined start and end boundaries, we also explore extended videos containing multiple irrelevant visual and action segments.

We evaluate the following three scenarios: (1) **Baseline**: Videos captured under standard clear conditions. (2) **With Visual Disturbance**: Videos that include background clutter or additional distracting objects, simulating more realistic visual environments. (3) **With Action Disturbance**: Videos where the demonstrated action is slightly altered (e.g., grasping a different object), introducing minor motion variations.

Our proposed GMS-SDTW method processes each long video to detect and maps subsequences corresponding to Pick and Place robot demonstration trajectory. As shown in Fig. C.1, action-based retrieval yields more precise results showing resilience to visual disturbances. Quantitative results, including mean Intersection over Union (mIoU) and accuracy at a threshold of 0.5, are presented in Tab. C.1. By focusing on action similarity, our system more accurately localizes the relevant segments while reducing sensitivity to irrelevant visual content. Overall, while visual-based mapping may suffer from real-world visual variations, action-based mapping remains robust and reliable.

D Additional Baseline Comparison via Visual Retrieval

We compare ImMimic-A with the current state-of-the-art retrieval-based method STRAP [14]. STRAP leverages a strong vision foundation model, DINOv2 [61] to embed each video frame and employs S-DTW to retrieve relevant subtrajectories. Following STRAP, each robot demonstration is first segmented into variable-length sub-trajectories using the low-level end-effector motion heuristic. We then extract DINOv2 features from agent-view videos for both human and robot data. Treating robot subtrajectories as a query, we apply S-DTW to locate matching subsequences in human videos. We cap the number of matches per query at $K = 500$ where K denotes the maximum number of matched segments per query. As shown in Tab. E.1, STRAP outperforms the Robot Only baseline, while ImMimic-A still achieves even higher performance. STRAP is designed for robot-to-robot transfer via retrieval-based matching and therefore does not explicitly address the domain

Setting	Robotiq		Ability	
	Pick and Place	Flip	Pick and Place	Flip
Robot Only	0.40	0.60	0.80	0.60
Co-Training	0.40	0.80	0.80	0.90
STRAP	0.50	0.60	0.90	0.90
ImMimic-A	**1.00**	**1.00**	**1.00**	**1.00**

Table E.1: Comparison of success rates between Robot Only, Co-Training, STRAP, and our ImMimic-A across two embodiments and two tasks, using 5 robot demonstrations and 100 human demonstrations.

Setting	Robotiq		Ability	
	Pick and Place	Flip	Pick and Place	Flip
ImMimic-A (β-dist)	0.90	0.90	0.90	**1.00**
ImMimic-A (linear)	**1.00**	**1.00**	**1.00**	**1.00**

Table E.2: Comparison between ImMimic-A (β-dist), which samples the MixUp ratio α from a β-distribution, and ImMimic-A (linear), which uses a linearly decreasing schedule for α. Success rates are reported across two embodiments and two tasks, using 5 robot demonstrations and 100 human demonstrations.

distribution gap present in human-to-robot transfer. Moreover, while STRAP employs a strong visual encoder for feature similarity, action information can offer more robust correspondence in the presence of a human-to-robot visual gap.

E Additional Experimental Results and Details

E.1 Domain Gap

Learning from human videos poses two critical gaps that often hinder policy transfer to robots: the *visual gap* and the *action gap* [3, 4]. The visual gap arises due to significant differences in appearance between humans and robots. The action gap stems from differences in kinematic constraints, motion dynamics, embodiment size, and task execution strategies.

Visual Gap. In Tab. E.4, we present sample demonstration clips highlighting how human and robot embodiments differ significantly in their visual observations. While a shared workspace setup can help reduce background-related visual discrepancies, notable appearance differences between human and robot demonstrations remain.

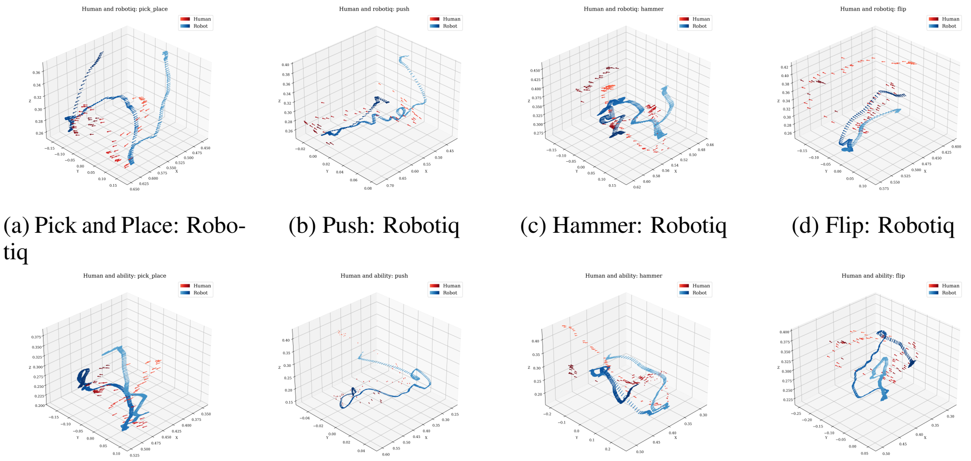
Embodiment	Pick and Place	Push	Hammer	Flip	AVG
Robotiq	0.031	0.067	0.085	0.083	0.066
FR	0.028	0.077	0.068	0.089	0.065
Allegro	0.063	0.065	0.089	0.094	0.078
Ability	0.047	0.056	0.091	0.106	0.075

Table E.3: Average action similarity across different embodiments and tasks. Grippers generally exhibit a smaller action gap compared to dexterous hands.

Action Gap. Fig. E.1 shows human demonstration trajectories overlaid with their corresponding teleoperated robot trajectories. Despite structural differences in design, retargeting aligns human and robot motions by emphasizing underlying physical similarities. Tab E.3 further quantifies human–robot action similarity.

E.2 Visualization of the Mapping

During MixUp, our mapping strategy ensures that interpolated demonstration pairs remain plausible to avoid generating infeasible demonstrations. Experimental results show that Random Mapping



(a) Pick and Place: Robotiq (b) Push: Robotiq (c) Hammer: Robotiq (d) Flip: Robotiq

(e) Pick and Place: Ability (f) Push: Ability (g) Hammer: Ability (h) Flip: Ability

Figure E.1: Visualization of sample trajectories pairs: the human retargeted trajectory and the corresponding robot demonstration trajectory. Arrows indicate orientation.

fails to improve performance, and ImMimic-V with its lower mapping quality, underperforms compared to ImMimic-A. We visualize an example of our action mapping at certain timesteps for the Robotiq Gripper and Ability Hand performing Pick and Place (Fig E.2). By sampling at different rates, we minimize the speed discrepancy between human and robot demonstrations to match their average durations. As shown in the figure, our mapping strategy effectively mpas observations and future states across embodiments, ensuring task-relevant consistency.

E.3 Visualization of Domain Flow

To illustrate how our methods adapts the across domains, we visualize the t-SNE [62] embeddings of human and robot conditions in Fig. E.3. Each point in the scatter plot represents a condition at a specific timestep from either human or the robot dataset. Under Vanilla Co-Training, human and robot data distributions remain clearly separated, highlighting the domain gap. This separation between the source (human) and target (robot) data indicates that, without explicit domain adaptation, the model cannot fully leverage human data for robot training. Similar to DLOW [16], which employs a continuous “domainness” variable to transition from source to target domains, ImMimic-A uses the mixing coefficient α to control how far each sample is adapted toward the robot domain.

E.4 MixUp with β-distribution

In several MixUp-based approaches [15, 41], α is sampled from a β-distribution to augment the data distribution. In Tab. E.2 we compare ImMimic-A (β-dist) to ImMimic-A (linear), where α is sampled directly from a β-distribution. Our results show that ImMimic-A (linear), which uses a linearly decreasing α schedule, still outperforms ImMimic-A (β-dist).

The results confirm that progressive MixUp scheduling enhances policy robustness across domains. Models trained with the linear α scheduler achieve better adaptation between human and robot distributions, leading to smoother trajectories and improved task success compared to the β-distributed variant. This demonstrates that controlled, gradual interpolation not only bridges the domain gap but also yields more stable and effective robot behaviors.

E.5 Success Rate Metrics

Success Rate. The four tasks are designed to evaluate various aspects of robotic manipulation. Each task includes specific disturbances to test robustness.

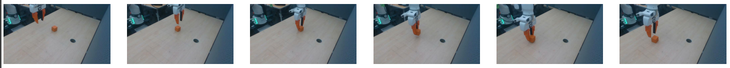
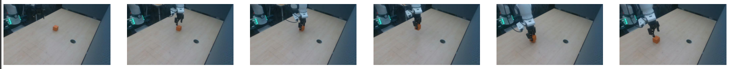
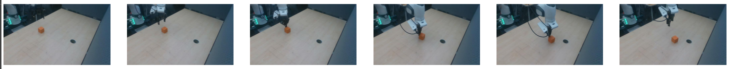
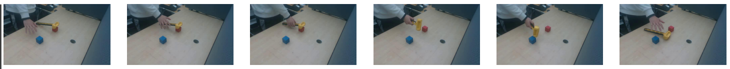
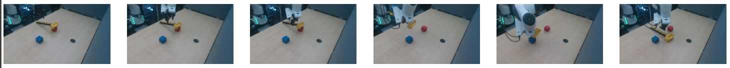
Task	Embodiment	Agent view
Pick and Place	Human	
	FR	
	Robotiq	
	Allegro	
	Ability	
Push	Human	
	FR	
	Robotiq	
	Allegro	
	Ability	
Hammer	Human	
	FR	
	Robotiq	
	Allegro	
	Ability	
Flip	Human	
	FR	
	Robotiq	
	Allegro	
	Ability	

Table E.4: Agent-view visualization for human and four different embodiments (FR, Robotiq, Allegro, Ability) performing four tasks (Pick and Place, Push, Hammer, Flip).

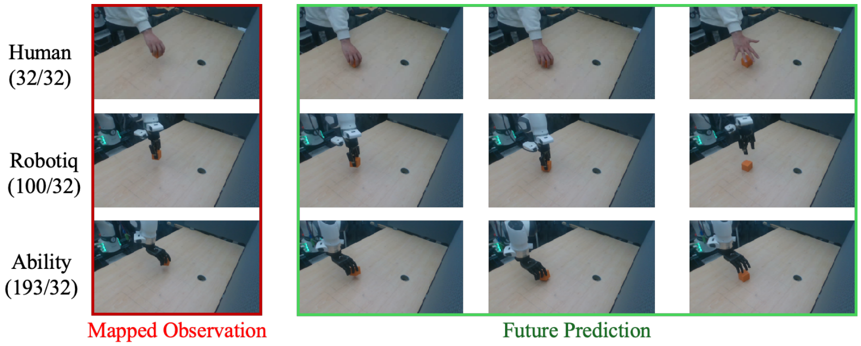


Figure E.2: An example of mapped pairs at the same timestep used for MixUp. As shown in Tab. A.2, we set sample rates γ (Human: 32/32, Robotiq: 100/32, Ability: 193/32) based on average durations to ensure consistent execution speed.

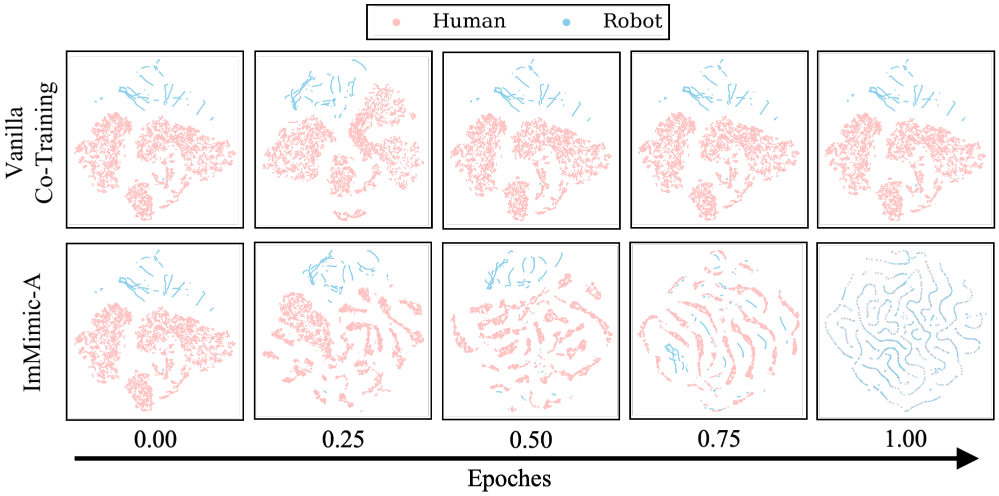


Figure E.3: t-SNE visualization of input conditions at each timestep from human and robot datasets during training. We compare ImMimic-A with Co-Training, showing that ImMimic-A generates a smooth domain flow for the human data, enabling effective domain adaptation.

1. Basic Object Manipulation. (1) **Pick and Place**: The robot must pick up a cube from a start position and place it at a designated target location. The initial position of the cube is roughly fixed but includes a random offset within the start area. This task evaluates the robot's ability to accurately grasp and relocate objects. The task is considered successful if the cube fully covers the target point. We consider the attempt successful if the cube fully covers the target point. (2) **Push**: The robot must push the object from the start position to the target region. This task primarily evaluates finger-free manipulation capabilities. Similar to the Pick and Place, a random offset is applied to the cube's initial position. The task is considered successful if the object reaches the target region after the push. *2. Tool-based Manipulation.* (1) **Hammer**: The robot must pick up a hammer and strike a target cube with its head. This task requires proper tool grasping and precise targeting. The hammer is initially placed on a cube, with its handle orientation randomly disturbed within a 45-degree range. The task is successful if the hammer's head touches the top surface of the target cube. (2) **Flip**: The robot must flip a bagel using a spatula after lifting it. This task emphasizes precise wrist control and rotational dexterity. The spatula is placed at an angle within 45 degrees, and the bagel is positioned randomly on different parts of its head. Success is defined as the bagel being flipped over.

Failure Cases. We summarize common failure modes observed across the four robotic embodiments.

Robotiq Gripper. In Push, Robotiq Gripper struggles to maintain a straight trajectory due to its thin fingertips, leading to unstable contact and frequent path corrections. In Flip, limited wrist articulation and low contact area make it difficult to control the spatula through the full rotation, resulting in intermittent slippage. Additionally, a structural gap above the fingertips can cause the gripper to grasp the spatula within this space, leading to an unstable grip. These issues are visually highlighted in Fig. 7(a,b), where Robotiq's fingertip geometry and palm gap contribute to contact instability and slippage.

Fin Ray Gripper. In Push, FR Gripper improves on Robotiq Gripper's stability but still lacks the fine precision of multi-fingered hands. In Flip, its limited wrist articulation leads to occasional loss of control during dynamic movements.

Allegro Hand. In Hammer, Allegro's relatively large hand size reduces its ability to generate sufficient lift force, making it difficult to wield heavier tools effectively. In Flip, the same size limitation, combined with weak grip force, often results in the spatula slipping before the motion completes. These failures are illustrated in Fig. 7(f,g), where the hand struggles to maintain stable tool contact during high-torque actions.

Ability Hand. In Pick and Place, the short thumb and limited wrist flexibility of the Panda arm often result in unstable grasps and frequent object drops. In Hammer, the same constraints hinder stable tool grasping and force transmission. As shown in Fig. 7(c,d), the shorter thumb may also contribute to misaligned grasps, especially when positional offsets are present.

Mechanical Design Insights. Analysis of failure cases reveals that no single hand design is universally optimal across all tasks. However, several general insights can inform more effective mechanical design of end-effector:

(1) Increase thumb length relative to other fingers to expand the acceptable grasping margin and reduce off-center spinning (supported by biological evidence [63]). A longer thumb increases the moment arm and provides greater contact redundancy, improving robustness when objects shift under load.

(2) Account for mounting and arm constraints. Most current end-effector mounts lack an additional wrist degree of freedom, limiting the ability to perform human-like reorientation. Introducing a swivel or universal joint at the mounting interface can restore this degree of freedom, enabling more favorable tool approaches without compromising the robot's kinematic reach.

(3) Enable firm, adaptive grasps by incorporating an adjustable thumb–finger aperture mechanism and compliant interface materials. A variable-spacing mechanism allows the hand to conform to different tool cross-sections, while soft, high-friction coatings compensate for local misalignments and absorb minor impacts, preventing slippage throughout the workspace.

E.6 Smoothness Metrics

Spectral Arc Length (SPARC) quantifies smoothness by measuring the arc length of the normalized magnitude spectrum of a trajectory's speed profile in the frequency domain [57], building on the original Spectral Arc Length (SAL) [64]. Given a speed profile s_t, the normalized spectrum is defined as:

$$\hat{S}(\omega) = \frac{S(\omega)}{S(0)} \tag{E.1}$$

The SAL metric is then computed as:

$$\text{SAL} \triangleq -\int_0^{\omega_c} \sqrt{\left(\frac{1}{\omega_c}\right)^2 + \left(\frac{d\,\hat{S}(\omega)}{d\omega}\right)^2}\, d\omega \tag{E.2}$$

SPARC improves upon SAL by adaptively selecting the cutoff frequency ω_c based on an amplitude threshold $\overline{S}$ and an upper frequency limit $\omega_c^{\max}$:

$$\omega_c \triangleq \min\left\{\omega_c^{\max},\, \min\left\{\omega \,\middle|\, \hat{S}(\gamma) < \overline{S},\, \forall\, \gamma > \omega\right\}\right\} \tag{E.3}$$

In our implementation, we apply zero-padding to the speed trajectory with a factor of $K = 4$, and set the parameters $\omega_c^{\max} = 15$, $\overline{S} = 0.05$. A higher SPARC score corresponds to a smoother trajectory. With the metric, we are able to show that our ImMimic improves the smoothness for the rollout policy to both Robot-Only and Co-Training.

E.7 Training Setup and Deployment Details

All models are trained for 300 epochs using an NVIDIA A40 GPU, with a batch size of 128. For deployment, we perform policy rollout with both inference and control running at 30 Hz on a desktop equipped with an NVIDIA RTX 4090 GPU. All robot sensors operate at 30 Hz, while the Zed and RealSense cameras stream at 30 FPS.

References

[1] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on robot learning*, pages 477–490. PMLR, 2022.

[2] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

[3] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[4] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

[5] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos. In *Conference on Robot Learning*, pages 654–665. PMLR, 2023.

[6] 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.

[7] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

[8] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *RSS*, 2022.

[9] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. 2022.

[10] N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada. Ditto: Demonstration imitation by trajectory transformation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7565–7572. IEEE, 2024.

[11] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song. Xskill: Cross embodiment skill discovery. In *Conference on robot learning*, pages 3536–3555. PMLR, 2023.

[12] 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.

[13] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. *arXiv preprint arXiv:2408.16944*, 2024.

[14] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. *arXiv preprint arXiv:2412.15182*, 2024.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. URL https://arxiv.org/abs/1710.09412.

[16] R. Gong, W. Li, Y. Chen, and L. V. Gool. Dlow: Domain flow for adaptation and generalization, 2019. URL https://arxiv.org/abs/1812.05418.

[17] R. Antunes, L. Lang, M. L. de Aguiar, T. Assis Dutra, and P. D. Gaspar. Design of fin ray effect soft robotic gripper for improved mechanical performance and adaptability: Numerical simulations and experimental validation. In *2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6, 2024. doi: 10.1109/MESA61532.2024.10704855.

[18] 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.

[19] K. Liu, C. Guan, Z. Jia, Z. Wu, X. Liu, T. Wang, S. Liang, P. Chen, P. Zhang, H. Song, et al. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. *arXiv e-prints*, pages arXiv–2409, 2024.

[20] G. Papagiannis, N. D. Palo, P. Vitiello, and E. Johns. R+x: Retrieval and execution from everyday human videos, 2025. URL https://arxiv.org/abs/2407.12957.

[21] V. Saxena, M. Bronars, N. R. Arachchige, K. Wang, W. C. Shin, S. Nasiriany, A. Mandlekar, and D. Xu. What matters in learning from large-scale datasets for robot manipulation. In *The Thirteenth International Conference on Learning Representations*.

[22] J. Shang, K. Schmeckpeper, B. B. May, M. V. Minniti, T. Kelestemur, D. Watkins, and L. Herlant. Theia: Distilling diverse vision foundation models for robot learning, 2024. URL https://arxiv.org/abs/2407.20179.

[23] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

[24] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection, 2024. URL https://arxiv.org/abs/2405.03666.

[25] K. Yu, Y. Han, Q. Wang, V. Saxena, D. Xu, and Y. Zhao. Mimictouch: Leveraging multimodal human tactile demonstrations for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2310.16917.

[26] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto. Open teach: A versatile teleoperation system for robotic manipulation, 2024. URL https://arxiv.org/abs/2403.07870.

[27] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.

[28] R. Li, H. Wang, and Z. Liu. Survey on mapping human hand motion to robotic hands for teleoperation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2647–2665, 2021.

[29] Z. Q. Chen, K. Van Wyk, Y.-W. Chao, W. Yang, A. Mousavian, A. Gupta, and D. Fox. Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *arXiv preprint 2209.14284*, 2022.

[30] Q. Chen, K. V. Wyk, Y.-W. Chao, W. Yang, A. Mousavian, A. Gupta, and D. Fox. Learning robust real-world dexterous grasping policies via implicit shape augmentation. In *Conference on Robot Learning*, 2023.

[31] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. In *IEEE International conference on robotics and automation (ICRA)*, 2023.

[32] Y. Han, Z. Chen, K. A. Williams, and H. Ravichandar. Learning prehensile dexterity by imitating and emulating state-only observations. *IEEE Robotics and Automation Letters*, 2024.

[33] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2021.

[34] Q. She, R. Hu, J. Xu, M. Liu, K. Xu, and H. Huang. Learning high-dof reaching-and-grasping via dynamic representation of gripper-object interaction. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4):97:1–97:14, 2022.

[35] Y. Chen, C. Wang, Y. Yang, and C. K. Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.

[36] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. In *Journal of Machine Learning Research*, volume 17, pages 2096–2030, 2016.

[38] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1989–1998, 2018.

[39] K. Gong, X. Liu, Y. Zhang, J. Feng, and D. Tao. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2477–2486, 2019.

[40] S. Abnar, M. Mutny, B. Scholkopf, and S. Bauer. Gift: Gradual domain adaptation via virtual intermediate domain generation. In *International Conference on Learning Representations (ICLR)*, 2021.

[41] Z. Xu, Y. Wang, Y. Chen, H. Jin, Y. Wang, and J. Shao. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.

[42] B. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2296–2306, 2017.

[43] J. Choi, S. Oh, J. Choi, and J. Lee. Visual domain-invariant policy learning for visual imitation. In *International Conference on Learning Representations (ICLR)*, 2023.

[44] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. In *Robotics: Science and Systems (RSS)*, 2018.

[45] A. D. Edwards and C. L. Isbell. Imitation learning from observations by minimizing inverse dynamics disagreement. In *International Conference on Machine Learning (ICML)*, pages 1745–1754, 2019.

[46] C. Tan, Y. Su, and J. Wang. Enhancing offline reinforcement learning via dynamics-aware mixup. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024. doi:10.1109/IJCNN60899.2024.10650992.

[47] T. Fickinger, M. Xu, J. B. Tenenbaum, and R. Urtasun. Cross-domain imitation learning via optimal transport. In *Conference on Robot Learning (CoRL)*, 2022.

[48] P. Zhang, X. Gao, Y. Wu, K. Liu, D. Wang, Z. Wang, B. Zhao, Y. Ding, and X. Li. Moma-kitchen: A 100k+ benchmark for affordance-grounded last-mile navigation in mobile manipulation. *arXiv preprint arXiv:2503.11081*, 2025.

[49] X. Gao, P. Zhang, D. Qu, D. Wang, Z. Wang, Y. Ding, and B. Zhao. Learning 2d invariant affordance knowledge for 3d affordance grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3095–3103, 2025.

[50] S. Choi, S. Han, W. Kim, J. Chae, W. Jung, and Y. Sung. Domain adaptive imitation learning with visual observation, 2023. URL `https://arxiv.org/abs/2312.00548`.

[51] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

[52] Y. Rong, T. Shiratori, and H. Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1749–1759, 2021.

[53] 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*, 2024.

[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

[55] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[56] Z. Cui, H. Pan, A. Iyer, S. Haldar, and L. Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024.

[57] Balasubramanian, Sivakumar and Melendez-Calderon, Alejandro and Roby-Brami, Agnes and Burdet, Etienne. On the analysis of movement smoothness. *Journal of NeuroEngineering and Rehabilitation*, 12(1):112, 2015. doi:10.1186/s12984-015-0090-9. URL `https://doi.org/10.1186/s12984-015-0090-9`.

[58] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[59] F. Ding, Y. Zhu, X. Wen, G. Liu, and C. X. Lu. Thermohands: A benchmark for 3d hand pose estimation from egocentric thermal images. *arXiv preprint arXiv:2403.09871*, 2024.

[60] Z. Liu and Y. Liu. Bridge the gap: From weak to full supervision for temporal action localization with pseudoformer, 2025. URL `https://arxiv.org/abs/2504.14860`.

[61] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL `https://arxiv.org/abs/2304.07193`.

[62] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[63] S. Almécija, J. B. Smaers, and W. L. Jungers. The evolution of human and ape hand proportions. *Nature communications*, 6(1):7717, 2015.

[64] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet. A Robust and Sensitive Metric for Quantifying Movement Smoothness. *IEEE Transactions on Biomedical Engineering*, 59 (8):2126–2136, 2012. doi:10.1109/TBME.2011.2179545.