

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

Anonymous Author(s)

Affiliation

Address

email

1 Appendix

2 A Demonstration Collection System

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

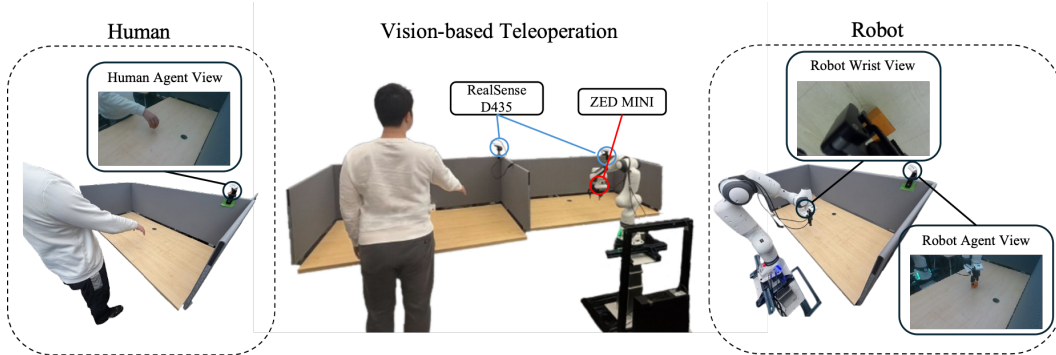


Figure A.1: Overall data collection system. (1) For human demonstrations, only the agent-view camera is used. (2) For robot demonstrations, both the agent-view and wrist-view cameras are used to enable precise control. (3) For teleoperation, a separate workspace is placed to the left of the robot, and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

8 A.1 Data Collection Throughput

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

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.

20 A.2 Sample Rate Normalization

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

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

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

36 A.3 Camera Calibration

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

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

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

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

$$\begin{bmatrix} \mathbf{p}_{\text{base}} \\ 1 \end{bmatrix} = \text{base}\mathbf{T}_{\text{cam}} \begin{bmatrix} \mathbf{p}_{\text{cam}} \\ 1 \end{bmatrix}. \quad (\text{A.2})$$

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

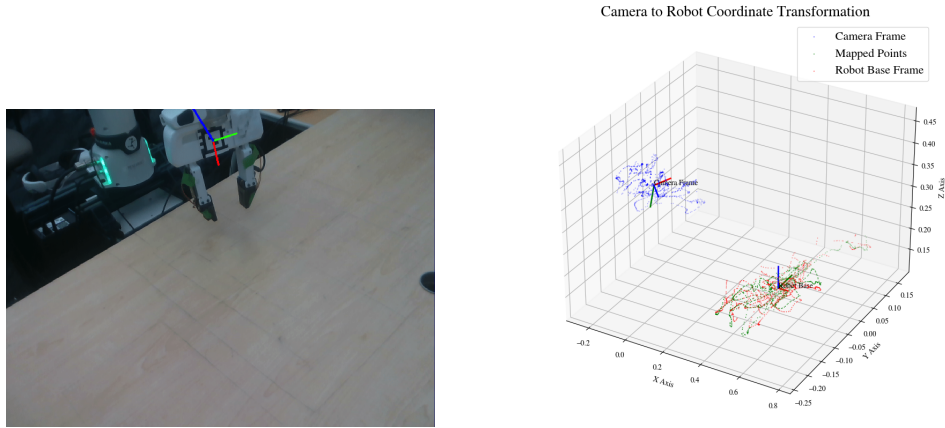
$$\text{base}\mathbf{T}_{\text{cam}} = \text{base}\mathbf{T}_{\text{tag}} (\text{tag}\mathbf{T}_{\text{cam}})^{-1}, \quad (\text{A.3})$$

51 Multiple such measurements enable us to refine (\mathbf{R}, \mathbf{t}) using a best-fit procedure. Given N pairs
 52 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
 53 (\mathbf{R}, \mathbf{t}) by minimizing:

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

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

54 We use a quaternion-based parameterization of \mathbf{R} to enforce $\text{SO}(3)$ constraint and solve the problem
 55 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.

56 B Retargeting

57 In both human videos processing and vision-based teleoperation for robot data collection, we per-
 58 form retargeting from human hand motion to robot actions. While the overall retargeting pipeline
 59 is shared across both settings, there are key differences. For human demonstration videos, we use
 60 offline retargeting based on RGB inputs and apply position retargeting, where absolute 3D joint
 61 positions are mapped to robot actions. For real-time vision-based teleoperation, we apply online
 62 retargeting that replaces wrist pose estimation with a more stable depth-based method and adopts
 63 vector retargeting [1], which aligns finger segment orientations rather than absolute positions for
 64 teleoperation. This section provides additional details on the retargeting process.

65 **Human Pose Estimation and Wrist Localization.** To estimate human hand pose, we use Me-
 66 diaPipe [2], a real-time pipeline that provides robust hand bounding boxes. Each cropped hand
 67 region is then passed to FrankMocap [3], which outputs shape and pose parameters for an SMPL-X
 68 model [4], resulting in accurate 3D coordinates for 21 knuckle joints in the local wrist frame.

69 To improve spatial accuracy, particularly important for teleoperation, we replace FrankMocap’s es-
 70 timated wrist translation with a wrist point derived from depth data captured by an RGBD camera.
 71 For wrist orientation, we apply the **Perspective-n-Point (PnP) algorithm** [5], solving:

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

72 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
 73 orientation matrix, t is the translation vector, and Π is the camera projection function. This yields a
 74 refined 6-DoF wrist pose that is consistent with the observed depth.

75 **Online Retargeting for Real-time Teleoperation.** For real-time teleoperation, we adopt vector re-
 76 targetting to ensure responsiveness and avoid kinematic singularities. Instead of matching absolute
 77 joint positions, we optimize finger orientations to follow the directions of human keypoint vectors.
 78 Given keypoint vectors \mathbf{v}_t^i from MediaPipe [2], we solve for the robot joint configuration by mini-
 79 mizing:

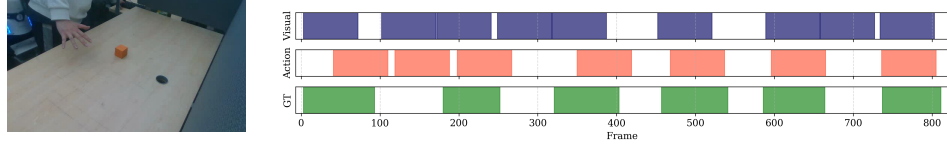
$$\min_{\mathbf{q}_t} \sum_{i=1}^N \|\alpha \mathbf{v}_t^i - \mathbf{R} f_i(\mathbf{q}_t)\|^2 + \beta \|\mathbf{q}_t - \mathbf{q}_{t-1}\|^2, \quad \text{s.t.} \quad \mathbf{q}_l \leq \mathbf{q}_t \leq \mathbf{q}_u, \quad (\text{B.2})$$

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

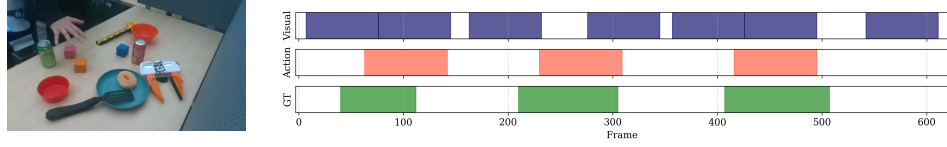
85 C Long Raw Human Video Retrieval

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

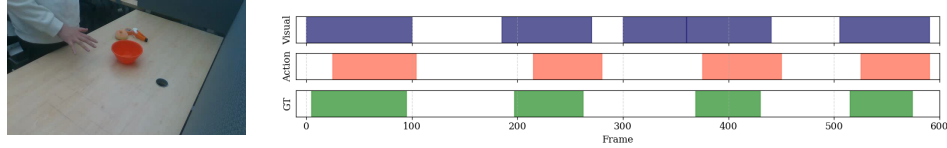
87 In our current setup, we use segmented human and robot demonstrations recorded in the same
 88 workspace while performing the same task. This controlled design minimizes the visual and ac-
 89 tion gap and simplifies the mapping process. In contrast, more practical scenario involves long,
 90 untrimmed human videos that include disturbances and task-irrelevant actions. In such cases, iden-
 91 tifying an accurate mapping strategy becomes even more critical. To extract useful segments from
 92 these raw videos, recent retrieval-based methods attempts to match human segments with corre-
 93 sponding robot behaviors, most often relying on visual features [6]. We formulate a retrieval task
 94 using long human videos, enabling a comparison between visual- and action-based mapping strate-
 95 gies to clarify which modality yields higher accuracy. To address this, we propose **Greedy Multi-**
 96 **Segment Subsequence DTW (GMS-SDTW)**, an extension of our current mapping algorithm.



(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.

97 **Overview of GMS-SDTW.** Given a long human tra-
 98 jectory $\mathbf{H} = \{\mathbf{h}_t\}_{t=1}^{T_h}$ that contains an unknown num-
 99 ber of action subsequences, and a single robot trajectory
 100 $\mathbf{R} = \{\mathbf{r}_s\}_{s=1}^{T_r}$, our goal is to identify *all* the human sub-
 101 sequences that best match the robot trajectory. We ex-
 102 tend classical Subsequence DTW (S-DTW) by scanning
 103 through \mathbf{H} using a sliding window method, greedily se-
 104 lecting mapped subsequences whose distance to the robot
 105 trajectory is below a predefined threshold ϵ . The slid-
 106 ing window length L is varied within a predefined range
 107 $L \in [L_{\min}, L_{\max}]$. The algorithm is presented in Alg. 1.

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

$$D(0, 0) = d(0, 0), \quad D(i, 1) = \sum_{k=1}^i d(k, 1), \quad D(1, j) = d(1, j) \quad (\text{C.1})$$

for $i = 1, \dots, T_h$ and $j = 1, \dots, T_r$.

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

111 The recursive update is follows the standard DTW formulation:

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

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

114 **Greedy search.** Starting at frame $t = 1$, we evaluate subsequence $\mathbf{H}_{t:t+L}$ for lengths $L \in$
 115 $[L_{\min}, L_{\max}]$ via S-DTW, get the subsequence with the minimum distance, and store it if $d^* < \epsilon$.

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

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 ;
1: window bounds L_{\min}, L_{\max} ; distance threshold ϵ
Ensure: Set \mathcal{P} of matched segments $(h_{\text{start}}, h_{\text{end}}, r_{\text{start}}, r_{\text{end}}, d)$
2: $\mathcal{P} \leftarrow \emptyset$; $t \leftarrow 1$
3: **while** $t + L_{\min} - 1 \leq T_h$ **do**
4: $d_{\text{best}} \leftarrow +\infty$
5: **for** $L = L_{\min}$ to $\min(L_{\max}, T_h - t + 1)$ **do**
6: $\mathbf{q} \leftarrow \mathbf{H}_{t:t+L-1}$
7: $(d, j_{\text{start}}, j_{\text{end}}, -) \leftarrow \text{S-DTW}(\mathbf{q}, \mathbf{R})$
8: **if** $d < d_{\text{best}}$ **then**
9: $d_{\text{best}} \leftarrow d$
10: $L^* \leftarrow L$
11: $j_{\text{start}}^* \leftarrow j_{\text{start}}$
12: $j_{\text{end}}^* \leftarrow j_{\text{end}}$
13: **end if**
14: **end for**
15: **if** $d_{\text{best}} < \epsilon$ **then**
16: Add $(t, t + L^* - 1, j_{\text{start}}^*, j_{\text{end}}^*, d_{\text{best}})$ to \mathcal{P}
17: $t \leftarrow t + L^*$ ▷ Skip matched subsequence
18: **else**
19: $t \leftarrow t + 1$
20: **end if**
21: **end while**
22: **return** \mathcal{P}

116 Stored subsequences are recorded as segments $(t, t + L^*, k^*, j^*)$ and the search resumes from
117 $t = t + L^* + 1$. Otherwise we increment $t \leftarrow t + 1$. The algorithm runs in $\mathcal{O}((L_{\max} - L_{\min}) T_r, T_h)$
118 time, and each robot frame is only assessed within the S-DTW dynamic-programming table.

119 **Complexity.** Each S-DTW distance has a time complexity of $\mathcal{O}(T_h T_r)$. With a linear scan over T
120 frames and at most $L_{\max} - L_{\min} + 1$ window lengths, the overall complexity is $\mathcal{O}((L_{\max} - L_{\min} +$
121 $1) T_r T_h)$, which is tractable in practice since $L_{\max} \ll T$.

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

123 As discussed in **Core Results**, action-based mapping tends to offer more robust performance than vi-
124 sual mapping. To further compare their performance, we propose a long raw video retrieval task [7]
125 as an intuitive way to assess the robustness of each mapping method under varying conditions. In
126 addition to segmented human demos with well-defined start and end boundaries, we also explore
127 extended videos containing multiple irrelevant visual and action segments.

128 We evaluate the following three scenarios: (1) **Baseline:** Videos captured under standard clear con-
129 ditions. (2) **With Visual Disturbance:** Videos that include background clutter or additional distract-
130 ing objects, simulating more realistic visual environments. (3) **With Action Disturbance:** Videos
131 where the demonstrated action is slightly altered (e.g., grasping a different object), introducing mi-
132 nor motion variations.

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

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.

141 D Additional Baseline Comparison via Visual Retrieval

142 We compare ImMimic-A with the current state-of-the-art retrieval-based method STRAP [6].
143 STRAP leverages a strong vision foundation model, DINOv2 [8] to embed each video frame and
144 employs S-DTW to retrieve relevant subtrajectories. Following STRAP, each robot demonstration is
145 first segmented into variable-length sub-trajectories using the low-level end-effector motion heuristic.
146 We then extract DINOv2 features from agent-view videos for both human and robot data. Treating
147 robot subtrajectories as a query, we apply S-DTW to locate matching subsequences in human
148 videos. We cap the number of matches per query at $K = 500$ where K denotes the maximum
149 number of matched segments per query. As shown in Tab. E.1, STRAP outperforms the Robot Only
150 baseline, while ImMimic-A still achieves even higher performance. STRAP is designed for robot-
151 to-robot transfer via retrieval-based matching and therefore does not explicitly address the domain
152 distribution gap present in human-to-robot transfer. Moreover, while STRAP employs a strong vi-
153 sual encoder for feature similarity, action information can offer more robust correspondence in the
154 presence of a human-to-robot visual gap.

155 E Additional Experimental Results and Details

156 E.1 Domain Gap

157 Learning from human videos poses two critical gaps that often hinder policy transfer to robots: the
158 *visual gap* and the *action gap* [9, 10]. The visual gap arises due to significant differences in appear-
159 ance between humans and robots. The action gap stems from differences in kinematic constraints,
160 motion dynamics, embodiment size, and task execution strategies.

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

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

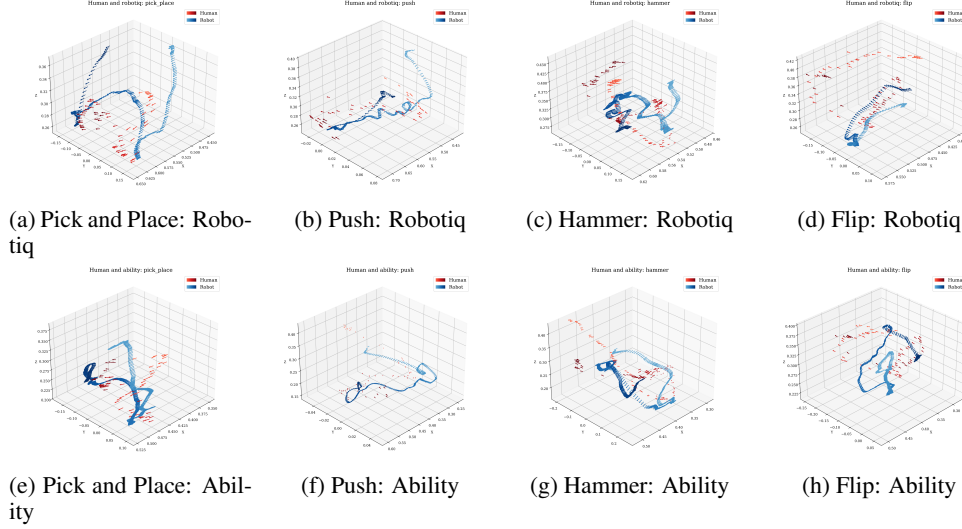


Figure E.1: Visualization of sample trajectories pairs: the human retargeted trajectory and the corresponding robot demonstration trajectory. Arrows indicate orientation.

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.

169 E.2 Visualization of the Mapping

170 During MixUp, our mapping strategy ensures that interpolated demonstration pairs remain plausible
171 to avoid generating infeasible demonstrations. Experimental results show that Random Mapping
172 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
173 Robotiq Gripper and Ability Hand performing Pick and Place (Fig E.2). By sampling at different
174 rates, we minimize the speed discrepancy between human and robot demonstrations to match their
175 average durations. As shown in the figure, our mapping strategy effectively maps observations and
176 future states across embodiments, ensuring task-relevant consistency.
177

178 E.3 Visualization of Domain Flow

179 To illustrate how our methods adapts the across domains, we visualize the t-SNE [11] embeddings
180 of human and robot conditions in Fig. E.3. Each point in the scatter plot represents a condition at
181 a specific timestep from either human or the robot dataset. Under Vanilla Co-Training, human and
182 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,
183 the model cannot fully leverage human data for robot training. Similar to DLOW [12], which employs a continuous “domainness” variable to transition from source to target domains, ImMimic-A
184 uses the mixing coefficient α to control how far each sample is adapted toward the robot domain.
185
186

187 E.4 MixUp with β -distribution

188 In several MixUp-based approaches [13, 14], α is sampled from a β -distribution to augment the
189 data distribution. In Tab. E.2 we compare ImMimic-A (β -dist) to ImMimic-A (linear), where α is






























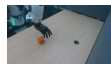





















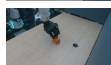
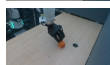
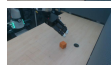
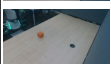
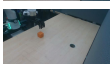
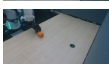
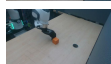
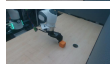
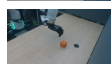












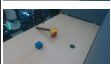
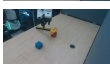
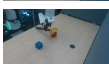
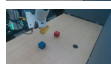
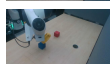
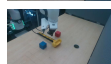



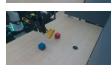
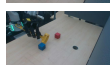
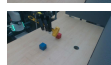
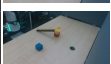
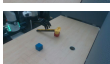
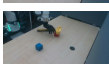









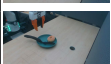

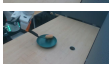
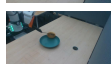

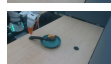










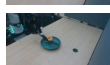



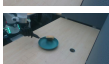



Task	Embodiment	Agent view					
Pick and Place	Human						
	FR						
	Robotiq						
	Allegro						
	Ability						
Push	Human						
	FR						
	Robotiq						
	Allegro						
	Ability						
Hammer	Human						
	FR						
	Robotiq						
	Allegro						
	Ability						
Flip	Human						
	FR						
	Robotiq					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

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.

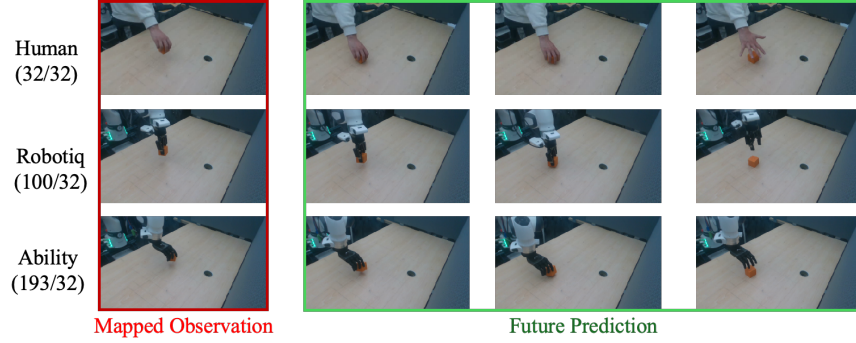


Figure E.2: An example of mapped pairs at the same timestep used for MixUp. As shown in Tab. A.2, we set sample rates γ (Human: 32/32, Robotiq: 100/32, Ability: 193/32) based on average durations to ensure consistent execution speed.

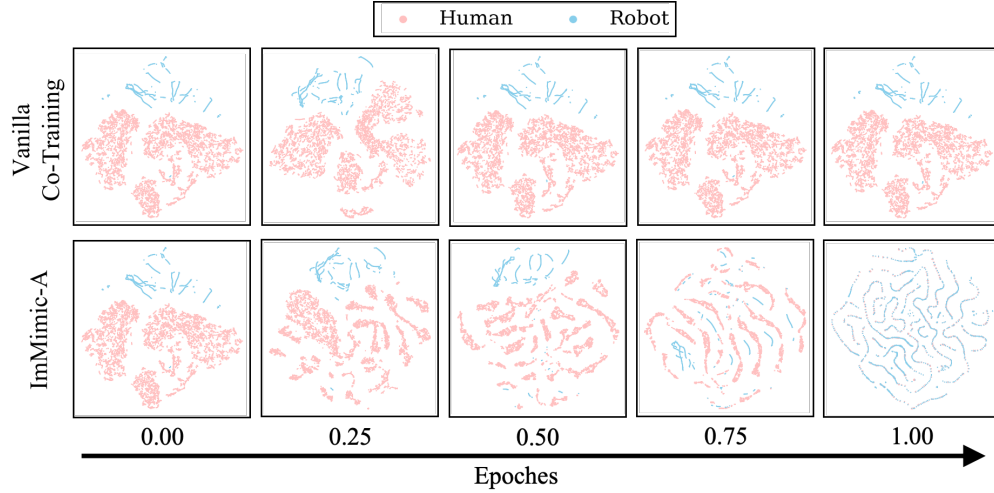


Figure E.3: t-SNE visualization of input conditions at each timestep from human and robot datasets during training. We compare ImMimic-A with Co-Training, showing that ImMimic-A generates a smooth domain flow for the human data, enabling effective domain adaptation.

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.

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

205 must push the object from the start position to the target region. This task primarily evaluates finger-
 206 free manipulation capabilities. Similar to the Pick and Place, a random offset is applied to the cube’s
 207 initial position. The task is considered successful if the object reaches the target region after the
 208 push. 2. *Tool-based Manipulation*. (1) **Hammer**: The robot must pick up a hammer and strike a
 209 target cube with its head. This task requires proper tool grasping and precise targeting. The hammer
 210 is initially placed on a cube, with its handle orientation randomly disturbed within a 45-degree range.
 211 The task is successful if the hammer’s head touches the top surface of the target cube. (2) **Flip**: The
 212 robot must flip a bagel using a spatula after lifting it. This task emphasizes precise wrist control and
 213 rotational dexterity. The spatula is placed at an angle within 45 degrees, and the bagel is positioned
 214 randomly on different parts of its head. Success is defined as the bagel being flipped over.

215 **Failure Cases.** We summarize common failure modes observed across the four robotic embodi-
 216 ments.

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

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

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

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

236 **Mechanical Design Insights.** Analysis of failure cases reveals that no single hand design is univer-
 237 sally optimal across all tasks. However, several general insights can inform more effective mechan-
 238 ical design of end-effector:

239 (1) Increase thumb length relative to other fingers to expand the acceptable grasping margin and
 240 reduce off-center spinning (supported by biological evidence [15]). A longer thumb increases the
 241 moment arm and provides greater contact redundancy, improving robustness when objects shift
 242 under load.

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

247 (3) Enable firm, adaptive grasps by incorporating an adjustable thumb–finger aperture mechanism
 248 and compliant interface materials. A variable-spacing mechanism allows the hand to conform to
 249 different tool cross-sections, while soft, high-friction coatings compensate for local misalignments
 250 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 [16], building on the original Spectral Arc Length (SAL) [17]. Given a speed profile s_t , the normalized spectrum is defined as:

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

The SAL metric is then computed as:

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

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

$$\omega_c \triangleq \min \left\{ \omega_c^{\max}, \min \left\{ \omega \mid \hat{S}(\gamma) < \bar{S}, \forall \gamma > \omega \right\} \right\} \quad (\text{E.3})$$

In our implementation, we apply zero-padding to the speed trajectory with a factor of $K = 4$, and set the parameters $\omega_c^{\max} = 15$, $\bar{S} = 0.05$. 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] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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>.

- 287 [8] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haz-
288 iza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li,
289 I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin,
290 and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL <https://arxiv.org/abs/2304.07193>.
291
- 292 [9] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay:
293 Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*,
294 2023.
- 295 [10] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu.
296 Egomimic: Scaling imitation learning via egocentric video, 2024. URL <https://arxiv.org/abs/2410.24221>.
297
- 298 [11] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning*
299 *research*, 9(11), 2008.
- 300 [12] R. Gong, W. Li, Y. Chen, and L. V. Gool. Dlow: Domain flow for adaptation and generalization,
301 2019. URL <https://arxiv.org/abs/1812.05418>.
- 302 [13] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk mini-
303 mization, 2018. URL <https://arxiv.org/abs/1710.09412>.
- 304 [14] Z. Xu, Y. Wang, Y. Chen, H. Jin, Y. Wang, and J. Shao. Adversarial domain adaptation with
305 domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34,
306 pages 6502–6509, 2020.
- 307 [15] S. Almécija, J. B. Smaers, and W. L. Jungers. The evolution of human and ape hand propor-
308 tions. *Nature communications*, 6(1):7717, 2015.
- 309 [16] Balasubramanian, Sivakumar and Melendez-Calderon, Alejandro and Roby-Brami, Agnes and
310 Burdet, Etienne. On the analysis of movement smoothness. *Journal of NeuroEngineering and*
311 *Rehabilitation*, 12(1):112, 2015. doi:10.1186/s12984-015-0090-9. URL [https://doi.org/](https://doi.org/10.1186/s12984-015-0090-9)
312 [10.1186/s12984-015-0090-9](https://doi.org/10.1186/s12984-015-0090-9).
- 313 [17] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet. A Robust and Sensitive Metric
314 for Quantifying Movement Smoothness. *IEEE Transactions on Biomedical Engineering*, 59
315 (8):2126–2136, 2012. doi:10.1109/TBME.2011.2179545.