FRAME-LEVEL TEMPORAL ALIGNMENT FOR HUMAN-TO-ROBOT VISUAL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferring visual representations pretrained on human videos to robot manipulation requires learning reliable correspondences between human and robot demonstrations. However, paired demonstrations can differ in execution rate and in the proportion of non-key frames that do not directly reflect task progress. Frames at the same relative timestamp may therefore represent different task stages, which can cause correspondence learning to fail. To address these issues, we propose Frame-Level Temporal Alignment (**FLTA**), a framework that uses two temporal priors to adapt visual encoders pretrained on human videos for robot manipulation. It learns shared task-progress representations without frame-level correspondence annotations, allowing frames at different relative temporal positions to match. A global progress prior combines normalized temporal positions with visual similarity to construct a soft correspondence target. A local temporal order prior penalizes backward transitions while allowing stays and varying forward rates to accommodate execution-rate differences. With ResNet-50 and ViT encoders, our method achieves relative improvements of 46.93% and 65.96%, respectively, over the best baselines in average simulation success rates and achieves higher task success rates on real-world manipulation tasks. These results also suggest that effective human–robot adaptation depends less on the number of parameters updated than on which parameters are selected. Our project page is available at `https://rtx5090ultra.github.io/FLTA-Project-Page/`.

1 INTRODUCTION

Vision-based robotic manipulation relies on visual representations that capture task-relevant state information for action prediction. Learning such representations solely from robot demonstrations is challenging because robot data are costly to collect and often cover a limited range of tasks. Human videos provide a scalable, complementary source of interaction data, as they are relatively inexpensive to obtain, widely available, and rich in object manipulation experience (Grauman et al., 2022; Bahl et al., 2023; Li et al., 2024; Srirama et al., 2024; Zhang et al., 2026). Pretraining on human videos can encode these experiences into transferable, task-relevant visual representations that support downstream policy learning and generalization beyond task-specific robot demonstrations (Nair et al., 2023; Radosavovic et al., 2023; Ma et al., 2023; Li et al., 2024).

However, differences in movement capabilities and appearance between humans and robots create a substantial domain gap between human videos and robot observations (Zhou et al., 2025; Ben-David et al., 2006; Ganin et al., 2016). As such, visual representations need to distinguish task progress because the downstream policy selects its next action based on the current observation (Nair et al., 2023; Srirama et al., 2024). HRAlign (Zhou et al., 2025), the state-of-the-art solution, uses paired human-robot videos for video-level contrastive alignment, learning shared task semantics to reduce this domain gap. However, frames sampled from paired human and robot videos may show different task stages despite sharing the same task instruction. Aligning their aggregated video representations does not explicitly establish which human and robot frames correspond to similar task progress.

This distinction is apparent for paired datasets such as RH20T (Fang et al., 2024), which provides human and robot demonstrations of the same task. As shown in Figure 1, these videos exhibit two temporal differences. First, the videos contain different proportions of non-key frames, which do not directly reflect task progress. As a result, the action starts at different relative positions in the two

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

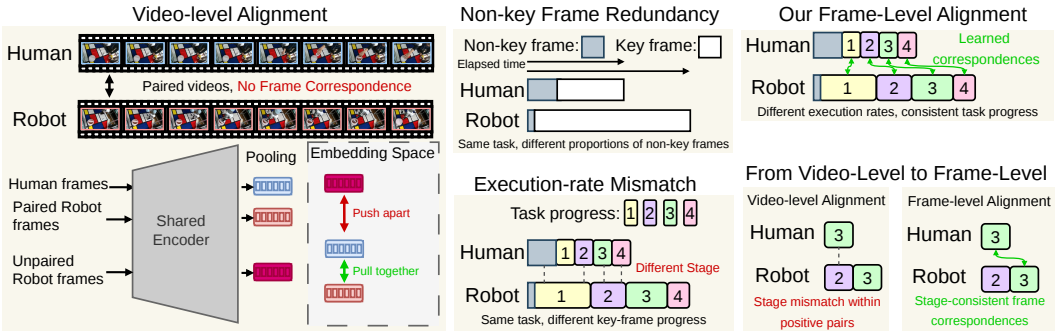


Figure 1: From video-level to frame-level alignment. Video-level alignment brings human and robot video representations closer without explicitly establishing frame-level correspondences. Differences in execution rates and the proportion of non-key frames mean that frames sampled at the same relative time may show different task stages. Our method addresses these temporal discrepancies by learning cross-domain frame correspondences without requiring frame-level annotations.

videos. Second, execution-rate mismatch changes the duration of each task stage. Frames sampled at the same relative time may show different task progress. Statistics across tasks and video pairs are provided in Appendix A.1. For example, sampled human non-key frames and robot key frames may contribute to a video-level positive pair. The aggregated video representations are pulled together even though the sampled frames show different task progress. Such stage mismatch may reduce the strength of the alignment signal used to learn task-progress representations, making it harder for downstream policies to select appropriate actions.

To address these issues, we propose a frame-level temporal alignment framework that adapts pre-trained visual encoders using paired human and robot videos, without frame-level correspondence annotations. As shown in Figure 1, we guide correspondence learning with the inductive bias that *human and robot demonstrations generally progress toward task completion over time* (Hadji et al., 2021; Zakka et al., 2022). First, we introduce a **global progress prior** D that combines normalized temporal positions with visual similarity to construct a soft correspondence target between human and robot frames. It allows each frame to match multiple candidate frames while penalizing large differences in normalized temporal position, rather than requiring same-position matches. Second, we introduce a **local temporal order prior** Q to guide the order of matches for consecutive sampled frames. As time progresses from the current human frame to the next sampled human frame, Q encourages the corresponding robot match to stay or move forward by penalizing backward transitions. This encourages temporal order while allowing for execution-rate differences. During adaptation, we update only the affine parameters of normalization layers, a restricted parameterization previously used for domain adaptation (Wang et al., 2021). This restricts the parameter search space during adaptation, with the aim of improving the visual representations for downstream robotic manipulation.

On AGNOSTOS (Zhou et al., 2026), our method achieves a 48.27% relative improvement in average success rate over HRAlign with R3M-RN50 (Nair et al., 2023) initialization. With D4R-ViT (Dasari et al., 2023) initialization, it achieves a 65.96% relative improvement over HRP (Srirama et al., 2024). With D4R-ViT initialization, it also improves success rates on all four real-world manipulation tasks compared with the Frozen and HRP baselines. Our experiments further show that effective human–robot adaptation does not require a large number of trainable parameters. Updating only 53.1K normalization-affine parameters achieves higher success rates than the late-adapter variant, despite using approximately $29.8\times$ fewer trainable parameters. Adaptation only takes approximately 13 hours on a single NVIDIA RTX 5090 GPU, without multi-GPU training. These results suggest that effective human–robot adaptation depends less on the number of parameters updated than on which parameters are selected.

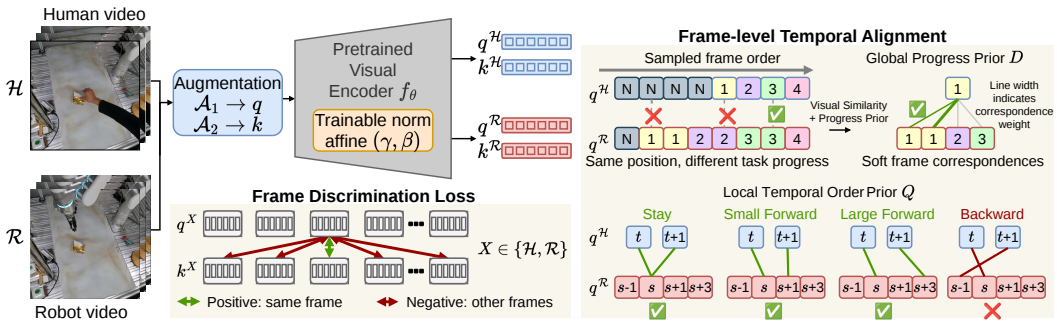


Figure 2: Overview of the proposed temporal alignment framework. Here, only the normalization-affine parameters of the visual encoder are updated. The auxiliary frame discrimination loss learns discriminative frame representations within each domain. The global progress prior D and visual similarity jointly construct soft correspondences. The local temporal order prior Q penalizes backward transitions while allowing stays and forward transitions of varying magnitudes. Numbered stages are shown only for illustration.

2 Human-to-Robot Domain Adaptation

We formulate the human-to-robot domain gap as a representation adaptation problem. For paired human and robot videos of the same task with different temporal structures, video-level alignment does not explicitly establish frame-level correspondences, as illustrated in Figure 1. We therefore adapt a pretrained visual encoder through frame-level temporal alignment without frame-level correspondence annotations. The adapted encoder is then frozen for downstream robot policy learning.

2.1 Problem Formulation

Let $\mathcal{D} = \{(L_\kappa, \mathcal{E}_\kappa)\}_{\kappa=1}^{K}$ denote a paired human–robot demonstration dataset, where K is the number of tasks. Here, L_κ is the instruction for task κ, and $\mathcal{E}_\kappa$ contains its human–robot video pairs. For each pair $(\mathcal{H}, \mathcal{R}) \in \mathcal{E}_\kappa$, $\mathcal{H}$ and $\mathcal{R}$ denote the human and robot demonstrations, respectively. The videos may differ in temporal structure, and no frame-level correspondence annotations are provided. For downstream policy learning, we train a policy to predict robot actions from robot visual observations. In the human-video pretraining paradigm, a visual encoder f_θ is first pretrained on human videos. The policy then takes the encoded representation $f_\theta(O)$, rather than the raw robot observation O, as its visual state input. To handle the human-robot domain gap, we adapt f_θ using $\mathcal{D}$. The adapted encoder provides visual representations better suited to robot observations O, helping the downstream policy predict the next action more reliably.

2.2 Framework Overview

As shown in Figure 2, our framework learns frame-level temporal correspondences using visual similarity, a global progress prior, and a local temporal order prior. An auxiliary frame discrimination loss supports cross-domain matching.

2.3 Frame-Level Temporal Alignment

Motivated by prior work on self-supervised temporal correspondence learning (Dwibedi et al., 2019; Haresh et al., 2021; Liu et al., 2022), we learn correspondences between paired human and robot videos without frame-level annotations. At each iteration, we uniformly sample $|\mathcal{T}|$ frames from each full video clip, including non-key frames, and retain their temporal order. Here, $\mathcal{T}$ denotes the sampled temporal index set, with the same number of frames $|\mathcal{T}| = 8$ sampled from each video by default. We index the sampled human and robot frames by $t, s \in \{0, \ldots, |\mathcal{T}|-1\}$, respectively, using their positions within the sampled sequences rather than their original video frame indices. Using these sampled sequences, we compute cross-domain visual similarities and guide correspondence learning with two temporal priors, supported by an auxiliary frame discrimination loss.

Visual Similarity. The encoder f_θ maps sampled human and robot frames to representations $q_t^{\mathcal{H}}$ and $q_s^{\mathcal{R}}$, respectively. We then construct a cross-domain visual similarity matrix S, where $S_{ts} = \mathrm{sim}(q_t^{\mathcal{H}}, q_s^{\mathcal{R}})$ is the cosine similarity between the two frame representations. We convert the similarity matrix S into the matching probability matrix P by applying a row-wise softmax, where P_{ts} is the predicted probability of matching human frame t to robot frame s.

Global Progress Prior D. As illustrated in Figure 2, human and robot frames at the same sampled position can represent different task progress. We therefore use soft correspondences across sampled positions, guided by both visual similarity and temporal position. We define $D_{ts} = |t-s|/(|\mathcal{T}|-1)$, so larger differences between the sampled frame indices t and s incur higher costs. Combining this prior with the visual matching cost $1 - S$ gives the soft correspondence target:

$$T = \mathrm{Sinkhorn}\left(\exp\left[-\frac{1 - S + \rho D}{\epsilon}\right]\right). \tag{1}$$

Sinkhorn normalization converts the combined costs into correspondence weights (Cuturi, 2013; Su & Wu, 2019). Each row of T assigns weights to candidate robot frames for one human frame. Here, ρ controls the weight of D, while larger ϵ values spread the weights more evenly across candidates.

Local Temporal Order Prior Q. We introduce a local temporal order prior Q to penalize backward transitions while allowing different execution rates. Here, P_{ts} gives the predicted probability of matching human frame t to robot frame s. For consecutive sampled human frames t and $t+1$, let s and s' denote their candidate correspondence positions in the robot sequence, respectively. Within the sampled robot sequence, s' may equal s, move forward to $s+1, s+2, \ldots$, or move backward to $s-1, s-2, \ldots$. We assign the following cost to each transition from s to s':

$$Q_{ss'} = \left(\frac{\max(s - s', 0)}{|\mathcal{T}| - 1}\right)^2. \tag{2}$$

This cost penalizes backward transitions ($s' < s$), with larger backward distances receiving higher penalties. Stays and forward transitions are not penalized by Q.

Bidirectional Alignment Objective. We train P to match the soft target T using cross-entropy and penalize backward transitions using Q. For human-to-robot alignment, the loss is:

$$\mathcal{L}_{\mathrm{TA}}^{\mathcal{H}\to\mathcal{R}} = \mathrm{CE}(T, P) + \mathcal{R}_Q(P). \tag{3}$$

Here, CE denotes the mean row-wise cross-entropy. We recompute T at each iteration but do not propagate gradients through it. The regularizer $\mathcal{R}_Q(P)$ discourages backward transitions in the predicted correspondences. We exchange the human and robot sequences to compute the reverse loss, then average the two directional losses to obtain $\mathcal{L}_{\mathrm{TA}}$. Detailed definitions are provided in Appendix A.3.

Frame Discrimination Loss. We use an auxiliary InfoNCE loss (Oord et al., 2018), denoted by $\mathcal{L}_{\mathrm{InfoNCE}}$, to learn discriminative frame representations for cross-domain alignment. Within each domain, augmented views of the same frame form positive pairs, while other frames in the same domain and batch serve as negatives. The full formulation is provided in Appendix A.4.

Overall Training Objective. The overall training objective is $\mathcal{L}_{\mathrm{total}} = \lambda_1 \mathcal{L}_{\mathrm{InfoNCE}} + \lambda_2 \mathcal{L}_{\mathrm{TA}}$, where λ_1 and λ_2 control the contributions of the two loss functions, respectively. We minimize this loss by updating only the affine parameters of the encoder's normalization layers; all other encoder parameters remain fixed.

3 Experiments

We aim to answer four questions. **(Q1)** Does frame-level temporal alignment improve the success rates of downstream manipulation policies in simulation and the real world? **(Q2)** Do the adapted visual representations achieve higher human-to-robot task-progress retrieval accuracy than HRP and HRAlign? **(Q3)** How do frame discrimination, the global progression prior, and local transition regularization contribute to performance, and how does our objective compare with alternatives? **(Q4)** How does performance vary across encoder initializations, adaptation strategies, and paired-data budgets? The results show improved manipulation success and task-progress retrieval accuracy,

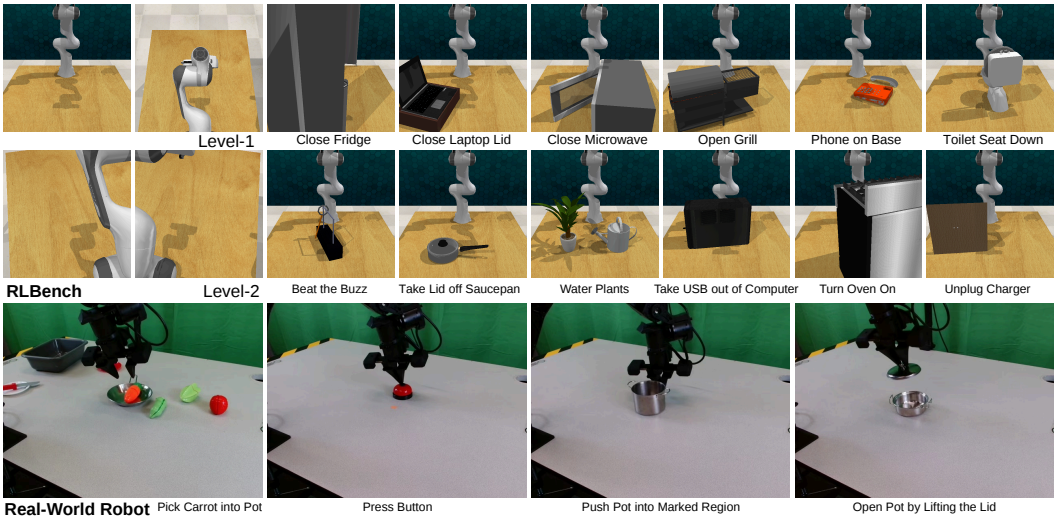


Figure 3: Simulated and real-world evaluation environments. Top: Level 1 and Level 2 unseen manipulation tasks from the AGNOSTOS benchmark in RLBench. Bottom: Four real-world tasks used on the Trossen WidowX AI Arm.

with ablations supporting the contributions of D and Q. Our method outperforms HRAlign under the same late-adapter configuration. For R3M-RN50, BN-affine fine-tuning achieves a higher success rate than our late-adapter variant with only 53.1K trainable encoder parameters.

Experimental Setups. Figure 3 shows the simulated and real-world evaluation environments.

In simulation, we evaluate cross-task zero-shot generalization using the AGNOSTOS benchmark (Zhou et al., 2026) in RLBench (James et al., 2020). Policies are trained on 18 task categories and evaluated on 23 unseen categories, with no overlap between the training and testing categories. Level-1 contains 13 tasks that share some object or action semantics with the training tasks. Level-2 contains 10 tasks with new object and action semantics, testing transfer beyond these familiar semantics. The complete task split is listed in Table 12 in the appendix. Following HRAlign's use of RVT for downstream manipulation (Zhou et al., 2025; Goyal et al., 2023), we adopt RVT-2 (Goyal et al., 2024) as our policy architecture. To evaluate the transferred representations, we insert the frozen visual encoder under evaluation into Stage 2 and reduce the number of policy attention layers in that stage from eight to one. Stage 1 remains unchanged.

For real-world evaluation, we use the Trossen WidowX AI arm for four tasks: *Pick Carrot into Pot*, *Press Button*, *Push Pot into Marked Region*, and *Open Pot by Lifting the Lid*. These tasks cover pick-and-place, button pressing, goal-directed pushing, and grasp-and-lift manipulation. We adopt ACT (Zhao et al., 2023) through the Trossen LeRobot training pipeline, replacing its visual encoder with the frozen encoder under evaluation. Each task uses 40 training demonstrations and 20 evaluation episodes. Environment and policy configurations are detailed in Appendix B.1.

Baselines. Our model is initialized based on two different encoder types: ViT (Dosovitskiy et al., 2020) and RN50 (ResNet-50) (He et al., 2016). In embodied learning, these two standard encoders represent different pretraining paths (Majumdar et al., 2023; Jing et al., 2023). R3M (Nair et al., 2023) uses an RN50 encoder to learn task progress over time from first-person human videos through temporal contrastive learning and visual-language alignment. In contrast, the ImageNet branch of D4R (Dasari et al., 2023) uses a ViT encoder to reconstruct masked patches in static images, but does not include explicit temporal supervision. Thus, the frozen R3M and D4R-ImageNet models were selected as our vanilla baselines. Our primary adaptation baselines are HRP-ImageNet (Srirama et al., 2024) and HRAlign (Zhou et al., 2025). HRP-ImageNet uses 150K human video clips from Ego4D (Grauman et al., 2022), without robot demonstrations for cross-domain alignment. Our method uses approximately 64K paired human–robot demonstrations from RH20T (Fang et al., 2024). HRAlign uses the same source dataset as our method but differs in data processing and

Table 1: Zero-shot manipulation success rates (%) on the 23 unseen AGNOSTOS tasks, with standard deviations in parentheses. Avg. denotes the average over all 23 tasks. Bold indicates the best result within each initialization group. Green indicates relative gains over HRP for D4R-ViT and HRAlign for the other initializations. DNCE denotes DecisionNCE. * denotes reproduced.

Encoder	Method	L1 ↑	L2 ↑	Avg. ↑	Encoder	Method	L1 ↑	L2 ↑	Avg. ↑
D4R-ViT	Frozen	8.31 (0.50)	11.60 (0.33)	9.74 (0.25)	DNCE-CLIP	Frozen	10.05 (1.24)	9.73 (1.32)	9.91 (1.14)
	+ HRP	10.56 (1.67)	12.13 (0.38)	11.25 (1.03)		+ HRAlign*	12.62 (2.06)	9.47 (1.05)	11.25 (1.58)
	+ Ours	**18.05 (0.52)**	**19.47 (0.50)**	**18.67 (0.49)**		**+ Ours**	**15.38 (0.44)**	**16.13 (1.00)**	**15.71 (0.22)**
		(+70.93%)	(+60.51%)	(+65.96%)			(+21.87%)	(+70.33%)	(+39.64%)
R3M-RN50	Frozen	12.62 (0.25)	11.73 (0.50)	12.23 (0.36)	AcTOL-CLIP	Frozen	11.28 (1.53)	11.47 (0.19)	11.36 (0.82)
	+ HRAlign	11.59 (1.62)	12.80 (0.33)	12.12 (0.78)		+ HRAlign*	11.90 (1.47)	12.27 (0.19)	12.06 (0.78)
	+ Ours	**16.82 (1.24)**	**19.47 (0.19)**	**17.97 (0.67)**		**+ Ours**	**13.23 (0.75)**	**21.73 (0.50)**	**16.93 (0.64)**
		(+45.13%)	(+52.11%)	(+48.27%)			(+11.18%)	(+77.10%)	(+40.38%)

fine-tuning strategy. To examine adaptation across different pretraining data and objectives, we additionally evaluate DecisionNCE (Li et al., 2024) and AcTOL (Zhang et al., 2026), both initialized from CLIP and further pretrained on EPIC-KITCHENS-100 (Damen et al., 2022), a human-video corpus with a shorter total duration than Ego4D used by R3M.

Implementation Details. We adapt the pretrained encoders using task-level human-robot video pairs from RH20T (Fang et al., 2024), retaining approximately 64K human–robot video pairs after data filtering. At each training iteration, we uniformly sample eight frames from each video. We set the loss weights to $\lambda_1 = 0.5$ and $\lambda_2 = 1$. We set $(\rho, \epsilon) = (0.25, 0.025)$ for R3M-RN50, DecisionNCE-CLIP, and AcTOL-CLIP, and $(0.5, 0.05)$ for D4R-ViT. Detailed training settings are provided in Appendix A.5.

3.1 Downstream Language-Conditioned Policy Evaluation

Simulation results. As shown in Table 1, our method achieves the highest average success rates on both L1 and L2 across all four encoder initializations. The overall relative gains are 65.96% over HRP with D4R-ViT and 48.27% over HRAlign with R3M-RN50. For DecisionNCE-CLIP and AcTOL-CLIP, the overall relative gains over HRAlign are 39.64% and 40.38%, respectively. Relative gains are larger on L2 for R3M-RN50, DecisionNCE-CLIP, and AcTOL-CLIP, whereas D4R-ViT shows a larger relative gain on L1. Per-task results in Table 9 show that lid opening makes the largest contribution to the improved L2 averages across all four encoder initializations. These gains across D4R, R3M, and CLIP-based initializations suggest that frame-level temporal alignment helps bridge the human–robot domain gap, improving zero-shot transfer of pretrained object and manipulation knowledge.

Real-world results. Given resource constraints, we select D4R-ViT for real-world evaluation, as it achieves the highest overall simulation success rate among the four initializations under our method. As shown in Table 2, our method improves success rates across all four tasks compared with Frozen and HRP. With D4R-ViT initialization and 40 demonstrations per task, it achieves an average success rate of 56.25%, a relative improvement of 73.08% over HRP. Our policy more reliably completes pick-and-place and pushing tasks under randomized object placements. In the pick-and-place task, it more frequently locates and picks carrots at unseen positions. In the pushing task, randomly positioned pots are more consistently moved into the target region. However, lid opening remains challenging for all methods. Our setup uses a single camera positioned diagonally behind the robot, without a complementary wrist-camera view. The limited spatial information and partial occlusion of the small lid make localization difficult, and grasps frequently fail.

Table 2: Real-world success rates using D4R-ViT initialization. T1–T4 denote pick–place, press, push, and open, respectively.

Method	T1	T2	T3	T4	Avg. ↑
Frozen	0.35	0.25	0.55	0.00	0.29
+ HRP	0.40	0.50	0.35	0.05	0.33
+ Ours	**0.65**	**0.60**	**0.85**	**0.15**	**0.56**

Table 3: Human-to-robot task-progress retrieval, reported as correct retrievals out of 100 queries per evaluation setting. Higher is better.

	In-distribution (RH20T)		Zero-shot (self-collected)	
Setting	D4R-ViT	R3M-RN50	D4R-ViT	R3M-RN50
Frozen	15/100	14/100	42/100	36/100
+ HRP / HRAlign	17/100	22/100	47/100	43/100
+ Ours	**39/100**	**43/100**	**56/100**	**61/100**

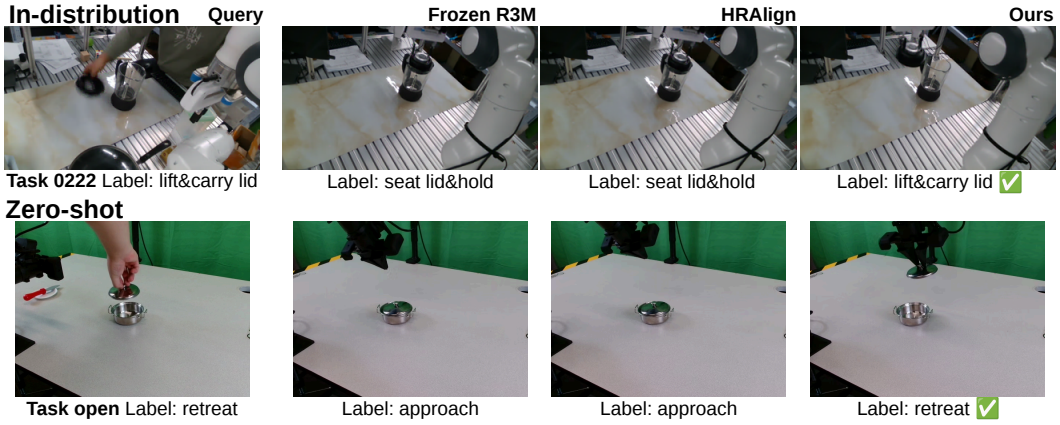


Figure 4: Human-to-robot task-progress retrieval examples from RH20T (top) and the self-collected zero-shot data (bottom). Each row shows a human query and the nearest robot frames retrieved by Frozen R3M, HRAlign, and our method. Green checks indicate correct retrievals.

3.2 Human-to-Robot Task-Progress Retrieval

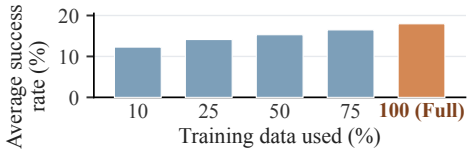
We evaluate whether human query frames retrieve robot frames at corresponding task progress under in-distribution and zero-shot settings. For in-distribution evaluation, we randomly select ten held-out human–robot video pairs from RH20T without training and randomly sample 100 human query frames across these pairs. For zero-shot evaluation, we use our self-collected data covering four manipulation tasks. For each self-collected task, queries from the five human videos retrieve their nearest-neighbor frames across all five robot videos. We randomly sample another 100 human query frames across this collection. We manually annotate task progress in both human and robot videos, using consistent stage definitions within each task. Each RH20T query retrieves its nearest-neighbor frame from the paired robot video based on visual similarity. A retrieval is considered correct when the query and retrieved frame share the same annotated progress. We report the number of correct retrievals out of 100 queries for each setting. As shown in Table 3, our method achieves the highest number of correct retrievals with both visual encoders in both settings. On RH20T, our method yields 22 more correct retrievals than HRP with D4R-ViT and 21 more than HRAlign with R3M-RN50. On the self-collected data, the corresponding gains are 9 and 18 correct retrievals out of 100 queries, respectively. These results provide evidence of task-progress correspondence across human and robot videos. Figure 4 illustrates retrieval examples from both settings, comparing our method with Frozen R3M and HRAlign. Annotation details and additional examples are provided in Appendix B.2.

3.3 Ablation Studies

Component Ablation. We evaluate four ablations by removing D, Q, both priors, or the auxiliary frame discrimination loss $\mathcal{L}_{\text{InfoNCE}}$. As shown in Table 4, the full model achieves the highest overall success rate with both encoders. Removing Q causes a larger performance drop than removing D. Removing both priors produces the largest drop among the component ablations. Removing

Table 4: Ablations and alternatives on the simulation. In the TCC and Soft-DTW variants, $\mathcal{L}_{\text{TA}}$ is replaced by the corresponding alignment loss.

Variant	D	Q	$\mathcal{L}_{\text{InfoNCE}}$	D4R-ViT			R3M-RN50		
				L1 ↑	L2 ↑	Avg. ↑	L1 ↑	L2 ↑	Avg. ↑
Full	✓	✓	✓	**18.05** (0.52)	**19.47** (0.50)	**18.67** (0.49)	**16.82** (1.24)	**19.47** (0.19)	**17.97** (0.67)
w/o $\mathcal{L}_{\text{InfoNCE}}$	✓	✓	–	16.15 (0.66)	17.27 (1.15)	16.64 (0.54)	16.31 (0.50)	17.73 (0.19)	16.93 (0.22)
w/o D	–	✓	✓	14.56 (0.77)	18.53 (0.38)	16.29 (0.43)	15.69 (1.26)	17.07 (0.82)	16.29 (0.91)
w/o Q	✓	–	✓	13.74 (0.77)	17.87 (0.50)	15.54 (0.46)	13.13 (0.38)	17.47 (0.68)	15.01 (0.36)
w/o D & Q	–	–	✓	11.18 (0.29)	15.20 (0.65)	12.93 (0.33)	11.79 (0.52)	15.47 (0.38)	13.39 (0.38)
Soft-DTW	–	–	✓	13.44 (0.52)	17.60 (0.57)	15.25 (0.22)	13.95 (1.02)	16.53 (0.33)	15.07 (0.59)
TCC	–	–	✓	13.13 (0.66)	14.27 (1.15)	13.62 (0.54)	12.41 (0.77)	13.07 (1.61)	12.70 (1.11)



(a) Data efficiency

Setting	Params. ↓	L1 ↑	L2 ↑	Avg. ↑
HRA + Ad.	1.58M	11.59	12.80	12.12
Ours + Ad.	1.58M	14.13	15.53	14.74
Ours + BN	**53.1K**	**16.82**	**19.47**	**17.97**

(b) Adaptation strategies

Figure 5: Data efficiency and adaptation strategies with R3M-RN50. (a) Overall average success rate across L1 and L2 versus the fraction of paired human–robot pre-training data. (b) Late-adapter (Ad.) versus BN-affine (BN) fine-tuning. Params.: trainable parameters; HRA: HRAlign. All success rates are in percent.

$\mathcal{L}_{\text{InfoNCE}}$ reduces success from 18.67% to 16.64% with D4R-ViT and from 17.97% to 16.93% with R3M-RN50. These ablations show that both temporal priors and frame discrimination contribute to downstream robot manipulation.

Alternative Alignment Objectives. We compare alignment objectives by replacing $\mathcal{L}_{\text{TA}}$ with TCC (Dwibedi et al., 2019) or Soft-DTW (Cuturi & Blondel, 2017), keeping all other settings unchanged. TCC forms correspondences through soft nearest-neighbor matching followed by a cycle-back step. In our experiments, we observe that visually noisy frames produce diffuse soft-NN weights, which can lead to incorrect cycle-back matches. Soft-DTW requires alignment paths to start and end at the sequence boundaries. When the boundary frames represent different task progress, this constraint can introduce incorrect correspondences. Our alignment objective achieves higher overall success rates than TCC and Soft-DTW with both encoders, as shown in Table 4.

3.4 Additional Analysis

Comparison of Encoder Adaptation Strategies. To evaluate whether our alignment method also benefits adapter-based fine-tuning, we adopt the late-adapter configuration used by HRAlign. We compare our complete alignment method with reproduced HRAlign using the same R3M initialization, downstream RVT-2 configuration, training data, hyperparameters, and number of training iterations. As shown in Figure 5(b), our method increases the overall success rate from 12.12% to 14.74% under late-adapter fine-tuning, corresponding to a relative improvement of 21.62%. This result indicates that the benefits of our alignment method extend beyond normalization-affine fine-tuning. With our alignment method, updating only 53.1K BN-affine parameters further increases the success rate from 14.74% to 17.97%, a relative improvement of 21.91%, while the late adapter uses approximately $29.8\times$ as many trainable parameters.

Data Efficiency of Human-Robot Pair Pre-training. We evaluate the effect of paired human–robot pre-training data size on downstream performance using R3M-RN50, reporting the overall average success rate across L1 and L2. We vary only the fraction of paired pre-training data while keeping other configurations fixed. As shown in Figure 5(a), the success rate increases monotonically with the fraction of training data, from 12.28% with 10% of the data to 17.97% with the full dataset. Using 75% of the data achieves a success rate of 16.48%. Given the improvement from 75% to

100% of the paired data, we expect further gains in downstream performance from expanding the pre-training dataset.

4 Related Work

R3M (Nair et al., 2023) learns reusable visual representations from human videos, while Decision-NCE (Li et al., 2024) captures task progression through sampled frame transitions. AcTOL (Zhang et al., 2026) further introduces temporal ordering and smoothness constraints. Complementary approaches emphasize spatial representations: MVP (Radosavovic et al., 2023) uses masked patch reconstruction, D4R (Dasari et al., 2023) examines pretraining data, and SlotMIM (Wen et al., 2025) emphasizes object-centric features. These methods provide useful initializations but do not explicitly establish human-robot video correspondences. For human-to-robot transfer, VRB (Bahl et al., 2023) predicts contact points and post-contact trajectories to guide manipulation, while HRP (Srirama et al., 2024) adapts visual representations using hand-object interaction cues. SWIM (Mendonca et al., 2023), DexWM (Goswami et al., 2025), and DreamDojo (Gao et al., 2026) instead learn proxy actions from human videos and adapt them to robot action spaces. We focus on cross-domain visual correspondences rather than directly transferring action distributions. Closest to our setting, HRAlign (Zhou et al., 2025) adapts pretrained models using paired human-robot videos, with task instructions guiding shared semantics for video-level alignment. In its limitations, HARP-VLA (Zhu et al., 2026) highlights execution-rate differences and redundant frames, emphasizing that robust transfer depends on reliable temporal alignment and shared correspondences. TCC (Dwibedi et al., 2019) uses soft nearest-neighbor matching and cycle consistency to encourage frames to return to their original temporal positions. Soft-DTW (Cuturi & Blondel, 2017) provides a differentiable objective over monotonic warping paths, permitting one-to-many and many-to-one matches, while LAV (Haresh et al., 2021) applies temporal alignment to representation learning. VAVA (Liu et al., 2022) combines optimal transport with temporal priors to accommodate action-order variations and handle unmatched background and redundant frames. Extending temporal alignment across embodiments, XIRL (Zakka et al., 2022) learns task-progress representations through temporal cycle consistency and uses them to define reinforcement learning rewards. Inspired by VAVA and XIRL, FLTA uses frame-level temporal correspondences to adapt pretrained visual encoders for downstream manipulation. We refine the video-level semantic alignment used by HRAlign through soft correspondence targets guided by D and local temporal order regularization through Q.

5 Conclusion and Limitations

We presented a frame-level temporal alignment framework for adapting pretrained visual encoders to robotic manipulation using paired human and robot videos. The global progress prior D and local temporal order prior Q guide correspondence learning, supported by an auxiliary frame discrimination loss. Adaptation updates only the affine parameters of normalization layers and requires no frame-level correspondence annotations. Experiments on AGNOSTOS show higher zero-shot success rates than HRP and HRAlign across four encoder initializations. Real-world experiments with D4R-ViT also show improved manipulation success compared with Frozen and HRP. Task-progress retrieval provides additional evidence of learned cross-domain frame correspondences.

Limitations of Real-World Transfer. Precise manipulation remains challenging with few robot demonstrations and a single camera view. With 40 demonstrations per task, the training data may not cover the range of object positions encountered during deployment. Small objects are also difficult to localize from a single external camera and can become occluded during manipulation, contributing to grasping failures. We plan to collect larger and more diverse robot demonstration datasets and incorporate complementary camera views into ACT.

Quality of Paired Videos. In some RH20T videos, shifts in camera viewpoint and objects moving out of frame make key parts of the manipulation difficult to observe. When key actions are not visible, visual similarity becomes less reliable for matching task progress across human and robot videos, making temporal alignment more difficult. Future work will explore more informative signals shared by human and robot demonstrations to reduce reliance on visual similarity for temporal alignment.

REFERENCES

Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 01–13. IEEE, 2023.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1183–1198. PMLR, 06–09 Nov 2023. URL `https://proceedings.mlr.press/v229/dasari23a.html`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 653–660. IEEE, 2024.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pp. 694–710. PMLR, 2023.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt2: Learning precise manipulation from few demonstrations. *RSS*, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano

Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, June 2022.

Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation learning via global temporal alignment and cycle-consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11068–11077, June 2021.

Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N. Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5548–5558, June 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707.

Ya Jing, Xuelin Zhu, Xingbin Liu, Qie Sima, Taozheng Yang, Yunhai Feng, and Tao Kong. Exploring visual pre-training for robot manipulation: Datasets, models and methods. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11390–11395. IEEE, 2023.

Jianxiong Li, Jinliang Zheng, Yinan Zheng, Liyuan Mao, Xiao Hu, Sijie Cheng, Haoyi Niu, Jihao Liu, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. DecisionNCE: Embodied multimodal representations via implicit preference learning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29461–29488. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/li24cr.html.

Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2181–2191, June 2022.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23301–23320. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ma23b.html.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? 2023.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pp. 416–426. PMLR, 2023.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. HRP: Human affordances for Robotic Pre-training. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.068.

Bing Su and Ying Wu. Learning distance for sequences by learning a ground metric. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6015–6025. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/su19b.html.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726(ICLR 2021)*, 2021.

Xin Wen, Bingchen Zhao, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. A data-centric revisit of pre-trained vision models for robot learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12143–12154. IEEE, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 537–546. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/zakka22a.html.

Zhizhen Zhang, Lei Zhu, Zhen Fang, Zi Huang, and Yadan Luo. Provable ordering and continuity in vision-language pretraining for generalizable embodied agents. *Advances in Neural Information Processing Systems*, 38:72370–72403, 2026.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22551–22561, June 2025.

Jiaming Zhou, Ke Ye, Jiayi Liu, Teli Ma, Zifan Wang, Ronghe Qiu, Kun-Yu Lin, Zhilin Zhao, and Junwei Liang. Exploring the limits of vision-language-action manipulation in cross-task generalization. *Advances in Neural Information Processing Systems*, 38:139899–139927, 2026.

Xiang Zhu, Puzhen Yuan, Yichen Liu, and Jianyu Chen. Harp-vla: Human-robot aligned representation learning for vision-language-action model. *arXiv preprint arXiv:2605.31234 (CoRL)*, 2026.

6 IMPACT STATEMENT

We employed large language models (LLMs) as tools for grammatical refinement, polishing the precision of word choice and the appropriateness of content expression. LLMs enhanced grammatical accuracy, textual clarity, and lexical selection. No aspects of the research design, implementation process, data analysis, or conceptual development relied on LLMs. This work studies the adaptation of visual representations for robotic manipulation using paired human-robot videos. The approach may support more effective reuse of human demonstrations for robot learning.

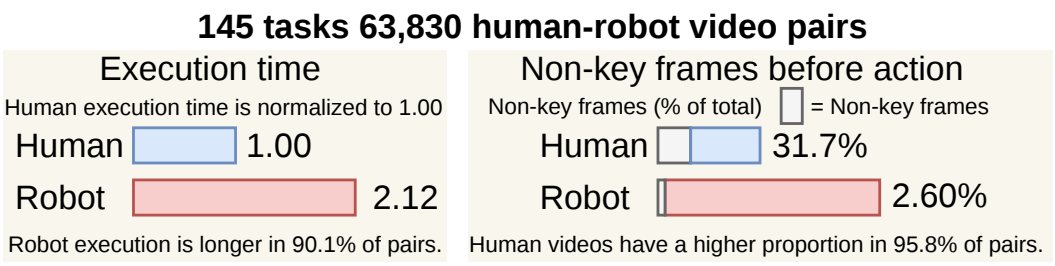


Figure 6: Temporal differences in paired human–robot videos. Left: median Robot-to-Human execution-time ratio, with Human time normalized to 1. Right: mean proportion of non-key frames before action execution.

Table 5: Core notation used in the proposed method.

Symbol	Meaning
$\mathcal{H}, \mathcal{R}$	Paired human and robot videos.
f_θ	Visual encoder with parameters θ; only normalization-affine parameters are updated during adaptation.
$\mathcal{A}_1, \mathcal{A}_2$	Augmentation operators for the two branches.
q_n^X, k_n^X	Representations of the same frame index n under the two augmentations, for domain $X \in \{\mathcal{H}, \mathcal{R}\}$.
$\mathcal{T}$, $\lvert\mathcal{T}\rvert$	Sampled temporal index set and number of sampled frames per video.
n, N	Frame-instance index and total number of sampled frames per domain.
t, s	Human frame index t and robot frame index s.
S	Cross-domain visual similarity matrix.
P	Predicted correspondence matrix.
D	Global progression cost matrix.
T	Soft correspondence target constructed from S and D.
Q	Local transition cost matrix.
ρ, ϵ	Global progression prior weight and Sinkhorn temperature, respectively.
$\mathcal{R}_Q(P)$	Regularization term that penalizes backward transitions under the predicted correspondences.
$\mathcal{L}_{\text{InfoNCE}}$	Intra-domain frame discrimination loss.
$\mathcal{L}_{\text{TA}}$	Cross-domain temporal alignment loss, averaged over both directions.
$\mathcal{L}_{\text{total}}$	Weighted sum of $\mathcal{L}_{\text{InfoNCE}}$ and $\mathcal{L}_{\text{TA}}$, with weights λ_1 and λ_2.

A Appendix

A.1 Temporal Differences in Paired Human–Robot Videos

Our cleaned RH20T dataset (Fang et al., 2024) contains 63,830 human–robot video pairs across 145 tasks. As shown in Figure 6, Robot execution is longer in 90.1% of pairs, with a median Robot-to-Human execution-time ratio of 2.12. The mean proportion of non-key frames before action execution is 31.7% for Human videos and 2.6% for Robot videos. Human videos have a higher proportion in 95.8% of pairs. Both differences hold across all 145 tasks when evaluated using within-task medians.

A.2 Notation

Table 5 lists the core notation used in the proposed method.

A.3 Details of the Bidirectional Alignment Objective

The following definitions refer to human-to-robot alignment, with directional superscripts omitted. Each row of T and P gives a distribution over robot frames.

Sinkhorn Normalization. We perform 20 Sinkhorn iterations, normalizing rows and then columns to sum to one. A final row normalization gives the soft correspondence target T.

Soft-target distillation. We minimize the mean row-wise cross-entropy to train P to match T:

$$\mathrm{CE}(T, P) = -\frac{1}{|\mathcal{T}|} \sum_{t=0}^{|\mathcal{T}|-1} \sum_{s=0}^{|\mathcal{T}|-1} T_{ts} \log P_{ts}. \tag{4}$$

We construct T from visual similarity and the global progression prior, and stop gradients through T.

Local transition regularization. For consecutive sampled human frames, we penalize backward transitions in their robot correspondences:

$$\mathcal{R}_Q(P) = \frac{1}{|\mathcal{T}| - 1} \sum_{t=0}^{|\mathcal{T}|-2} P_{t,:}\, Q\, (P_{t+1,:})^\top. \tag{5}$$

The two rows of P weight the transition costs in Q by their predicted probabilities. Q penalizes backward transitions but not stays or forward transitions.

Bidirectional averaging. We exchange the human and robot sequences to compute the reverse objective, then average the two directional losses:

$$\mathcal{L}_{\mathrm{TA}} = \frac{1}{2} \left(\mathcal{L}_{\mathrm{TA}}^{\mathcal{H}\to\mathcal{R}} + \mathcal{L}_{\mathrm{TA}}^{\mathcal{R}\to\mathcal{H}} \right). \tag{6}$$

A.4 Frame Discrimination Loss

Contrastive Objective. We introduce InfoNCE (Oord et al., 2018) to learn representations that can distinguish between different frames. We therefore applied data augmentation techniques before the visual encoder f_θ maps the frame images. For each frame x_n^X from domain $X \in \{\mathcal{H}, \mathcal{R}\}$, two augmentation operators $\mathcal{A}_1$ and $\mathcal{A}_2$ produce representations $q_n^X = f_\theta(\mathcal{A}_1(x_n^X))$ and $k_n^X = f_\theta(\mathcal{A}_2(x_n^X))$, where n indexes these sampled frames across videos. Let N denote the total number of sampled frame instances per domain, expressed as $N = (\text{number of video pairs}) \times |\mathcal{T}|$. We average the directional InfoNCE losses over both augmentation directions and both domains:

$$\mathcal{L}_{\mathrm{InfoNCE}} = \frac{1}{4} \sum_{X\in\{\mathcal{H},\mathcal{R}\}} \left[\ell_{\mathrm{NCE}}(q^X, k^X) + \ell_{\mathrm{NCE}}(k^X, q^X) \right]. \tag{7}$$

Here, ℓ_{NCE} denotes the directional InfoNCE loss (Oord et al., 2018), averaged over the N frames. Different augmented branches of the same frame form positive pairs, while representations from other frames serve as negative samples.

A.5 Pretraining Details

The pretraining settings and hyperparameters provided by HRAlign (Zhou et al., 2025) and HRP (Srirama et al., 2024) serve as a reference for us. However, due to limited computational resources, all of our pretraining was performed on a single NVIDIA RTX 5090, and our dataset utilized RH20T, a task-level dataset of human-robot pairs (Fang et al., 2024). With approximately 64K human–robot video pairs, pretraining on the ViT encoder took approximately 14.10 hours, while pretraining on the RN50 encoder took 12.97 hours. Details of the pretraining hyperparameters are shown in Table 6 and Table 7. We examine the effect of the number of sampled frames in Appendix A.5.1. The loss weights are set to $\lambda_1 = 0.5$ and $\lambda_2 = 1$ for both encoders. We set $(\rho, \epsilon) = (0.25, 0.025)$ for R3M-RN50, DecisionNCE-CLIP, and AcTOL-CLIP, and $(0.5, 0.05)$ for D4R-ViT. And, appearance perturbation includes color jitter, grayscale, blur, and erasing; all inputs are normalized.

A.5.1 Number of Sampled Frames

We evaluate the effect of the number of frames sampled per video during paired human–robot pretraining using R3M-RN50. We vary the number from three to ten and report the overall average

Table 6: Pretraining configurations for ViT and RN50.

Configuration	ViT	RN50
Training iterations	40,000	40,000
Optimizer	AdamW	AdamW
Learning rate	7.5×10^{-5}	1.0×10^{-4}
Frames per domain	160	160
Sampled frames	8	8
Weight decay	0	0
Loss weights (λ_1, λ_2)	(0.5,1)	(0.5,1)
Global-prior weight ρ	0.5	0.25
Sinkhorn temperature ϵ	0.05	0.025
Augmentation $\mathcal{A}_1/\mathcal{A}_2$	Resize / Appearance	Resize / Appearance

Table 7: Pretraining configurations for DecisionNCE-CLIP and AcTOL-CLIP.

Configuration	DecisionNCE-CLIP	AcTOL-CLIP
Training iterations	40,000	40,000
Optimizer	AdamW	AdamW
Learning rate	1.0×10^{-4}	1.0×10^{-4}
Frames per domain	160	160
Sampled frames	8	8
Weight decay	0	0
Loss weights (λ_1, λ_2)	$(0.5, 1)$	$(0.5, 1)$
Global-prior weight ρ	0.25	0.25
Sinkhorn temperature ϵ	0.025	0.025
Augmentation $\mathcal{A}_1/\mathcal{A}_2$	Resize / Appearance	Resize / Appearance

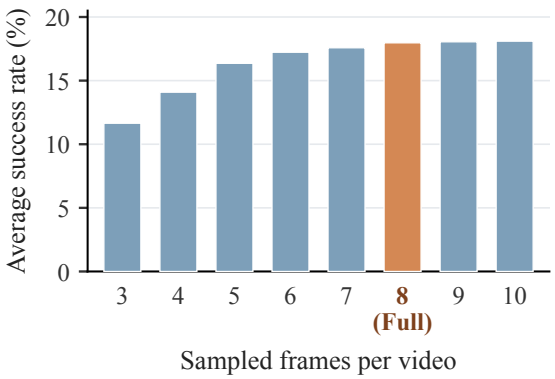


Figure 7: Effect of the number of sampled frames per video on the overall average success rate across L1 and L2 simulation tasks using R3M-RN50. Full denotes the default eight-frame configuration.

success rate across L1 and L2 simulation tasks. As shown in Figure 7, the success rate increases from 11.65% with three frames to 17.97% with eight frames. Sampling nine and ten frames yields success rates of 18.05% and 18.10%, respectively. These correspond to gains of only 0.08 and 0.13 percentage points over the eight-frame configuration. In our single-GPU setup, the eight-frame configuration already approaches the GPU memory limit. Sampling nine or ten frames per video requires reducing the batch size from 20 to 16. Under these configurations, adaptation takes 15–17 hours, compared with 12.97 hours for eight frames. Given the limited performance gains, we adopt eight frames per video as our default configuration, providing a practical trade-off between downstream performance, GPU memory requirements, and adaptation time.

Table 8: Downstream policy training configurations.

Simulation configurations

Configuration	ViT	RN50
Policy architecture	Modified RVT2	
Stage 1 attention layers	8	
Stage 2 attention layers	1	
Attention dimension	512	
Optimizer	LAMB	
Base learning rate	5×10^{-5}	
Weight decay	1×10^{-4}	
LR schedule	Cosine	
Warm-up steps	500	
Batch size/GPU	3	
Training epochs	1	
Total training iterations	25,000	

Real-world configurations

Configuration	ViT
Policy architecture	ACT
Transformer dimension	512
Attention heads	8
Activation	ReLU
Optimizer	AdamW
Learning rate	1×10^{-5}
Weight decay	1×10^{-4}
Training steps	8K
Batch size	16
Proprioception	7-D joint state
Demonstrations	40
Image augmentation	None

B Evaluation Details

B.1 Downstream Policy Architectures and Hyperparameters

RLBench 3D Manipulation Policy. Table 8 summarizes the simulation training configurations. The original RVT-2 architecture consists of two stages. The first stage employs an independent 8-layer attention network for coarse global localization. The second stage employs another independent 8-layer attention network for local, precise prediction. The 8-layer attention network in the second stage is then used to predict the end-effector's position, rotation, gripper state, and collision control state. In contrast, RVT1 has only 1 stage, using a single 8-layer attention network to directly predict the final gripper action from the complete scene, without a local zoom-in or local refinement stage. To evaluate the transfer of pretrained visual representations, we follow HRAlign's lightweight policy design. We extract features from rendered virtual views using the frozen visual encoder and process them with a single policy attention layer. We extended this design to RVT2, but modified only the second stage. We retain the original eight attention layers in the first stage for coarse global localization. The policy was trained on 18 standard RLBench training tasks, using 100 expert demonstrations per task for a total of 1,800 training trajectories. To reduce computational cost and assess the contribution of pre-trained visual encoders under a limited downstream training budget, we train both the ViT-based and RN50-based policies for 25,000 iterations, corresponding to one epoch in our training configuration. We use a global batch size of 3, compared with 192 in the original RVT-2. We evaluate each trained policy checkpoint in three separate runs. Each run uses 25 randomly selected test demonstrations per task, with success determined by the task-defined criteria. We report the mean success rates for Level-1, Level-2, and all 23 tasks, together with the standard deviations across the three evaluation runs.

Real-World Behavior Cloning Policy. We adopt ACT as the downstream policy architecture, following the official Trossen–LeRobot training pipeline for the WidowX AI. We replace its default ResNet-18 visual encoder with the encoder under evaluation while keeping all other policy and training configurations unchanged. The substituted encoder remains frozen during policy training, allowing us to isolate the effect of its learned visual representations rather than that of end-to-end adaptation. Detailed configurations are provided in Table 8.

B.2 Human-to-Robot Task-Progress Retrieval

We manually annotate task stages in the human and robot videos used for the in-distribution and zero-shot retrieval evaluations. Stage definitions are task-specific and are applied consistently to the human and robot videos of the same task. Frames at shared stage boundaries are assigned to the later stage. The complete frame ranges, stage names and other details are provided in Tables 10 and 11. For each evaluation setting, we randomly sample 100 human query frames. Each RH20T query retrieves its nearest-neighbor frame from the paired robot video based on visual similarity. For each self-collected task, queries from all five human videos retrieve frames from a shared pool of the five

Table 9: Per-task success rates (%) on the 23 unseen AGNOSTOS tasks, averaged over three evaluation runs. Avg. reports the average success rate for each level, and Std. its standard deviation across runs. DNCE denotes DecisionNCE. * denotes reproduced.

(a) Level-1 tasks

Encoder	Method	Toilet	Knife	Fridge	Micro.	Laptop	Phone	Seat	Lamp Off	Lamp On	Book	Umb.	Grill	Bin	Avg.	Std.
D4R-ViT	Frozen	0	0	16.0	18.7	14.7	0	56.0	0	0	0	0	2.7	0	8.31	0.50
	+ HRP	0	1.3	22.7	10.7	32.0	0	45.3	2.67	2.67	0	1.3	18.7	0	10.56	1.67
	+ Ours	0	0	64.00	54.67	28.00	0	73.33	0	0	0	1.33	12.00	1.33	**18.05**	0.52
R3M-RN50	Frozen	0	1.3	29.3	32.0	13.3	0	73.3	1.3	1.3	1.3	0	10.7	0	12.62	0.25
	+ HRAlign	0	0	37.3	14.7	24.0	0	65.3	6.7	1.3	0	0	1.3	0	11.59	1.62
	+ Ours	0	6.67	64.00	33.33	46.67	0	61.33	1.33	0	0	0	5.33	0	**16.82**	1.24
DNCE-CLIP	Frozen	0.00	1.33	26.67	12.00	24.00	0.00	54.67	1.33	0.00	0.00	1.33	6.67	2.67	10.05	1.24
	+ HRAlign*	0.00	1.33	44.00	6.67	33.33	1.33	60.00	0.00	2.67	0.00	1.33	4.00	9.33	12.62	2.06
	+ Ours	0.00	1.33	53.33	22.67	36.00	5.33	78.67	0.00	0.00	0.00	0.00	0.00	2.67	**15.38**	0.44
AcTOL-CLIP	Frozen	0.00	1.33	28.00	21.33	29.33	0.00	60.00	0.00	0.00	0.00	0.00	2.67	4.00	11.28	1.53
	+ HRAlign*	0.00	2.67	36.00	28.00	13.33	0.00	64.00	0.00	0.00	0.00	0.00	1.33	9.33	11.90	1.47
	+ Ours	0.00	1.33	49.33	22.67	41.33	4.00	50.67	0.00	0.00	1.33	0.00	1.33	0.00	**13.23**	0.75

(b) Level-2 tasks

Encoder	Method	USB	Lid	Plate	Ball	Scoop	Rope	Oven	Buzz	Plants	Charger	Avg.	Std.
D4R-ViT	Frozen	94.7	8.0	0	0	0	0	10.7	0	2.7	0	11.60	0.33
	+ HRP	100.0	0	0	0	0	0	10.7	4.0	5.3	1.3	12.13	0.38
	+ Ours	93.33	85.33	0	0	0	0	5.33	10.67	0	0	**19.47**	0.50
R3M-RN50	Frozen	100.0	0	0	0	0	0	6.7	9.3	0	1.3	11.73	0.50
	+ HRAlign	93.3	0	0	0	0	0	25.3	5.3	4.0	0	12.80	0.33
	+ Ours	94.67	84.00	0	0	0	0	6.67	6.67	2.67	0	**19.47**	0.19
DNCE-CLIP	Frozen	78.67	2.67	0.00	0.00	0.00	0.00	6.67	6.67	2.67	0.00	9.73	1.32
	+ HRAlign*	73.33	5.33	0.00	0.00	0.00	0.00	10.67	2.67	0.00	2.67	9.47	1.05
	+ Ours	68.00	69.33	4.00	2.67	0.00	0.00	9.33	2.67	1.33	4.00	**16.13**	1.00
AcTOL-CLIP	Frozen	92.00	0.00	0.00	0.00	0.00	0.00	5.33	8.00	1.33	8.00	11.47	0.19
	+ HRAlign*	94.67	2.67	1.33	0.00	0.00	0.00	14.67	6.67	1.33	1.33	12.27	0.19
	+ Ours	98.67	94.67	2.67	0.00	0.00	0.00	14.67	5.33	1.33	0.00	**21.73**	0.50

robot videos. A retrieval is counted as correct only when both the query and retrieved frame belong to the same annotated stage. We report the number of correct retrievals out of 100 queries in Table 3.

Figures 8 and 9 provide additional task-progress retrieval examples from RH20T and the self-collected zero-shot data, respectively. In these selected examples, our method retrieves robot frames from progress segments corresponding to those of the human queries. For the *press* task, we annotate two progress segments: *approach and press* and *release and retreat*. In the example shown in Figure 9, the human query belongs to the *approach and press* segment. Frozen R3M and HRAlign retrieve robot frames outside the annotated action interval, whereas our method retrieves a frame from the corresponding *approach and press* segment. Together with the quantitative results in Table 3, these examples provide evidence of task-progress correspondence across human and robot videos and complement the downstream performance results.

B.3 Simulation Environment

RLBench (James et al., 2020) is a simulated manipulation environment built around the Franka Panda robot and provides multi-view RGB-D observations. Following the widely adopted 18-task protocol established by PerAct (Shridhar et al., 2023) and subsequently used by RVT (Goyal et al., 2023) and RVT-2 (Goyal et al., 2024), a single language-conditioned policy is trained and evaluated on the same 18 task categories, comprising 249 task variations. This protocol primarily evaluates generalization to variations within seen task categories, rather than zero-shot generalization to entirely unseen tasks. We additionally follow the AGNOSTOS benchmark introduced with X-ICM (Zhou et al., 2026). AGNOSTOS retains the standard 18 RLBench tasks for downstream

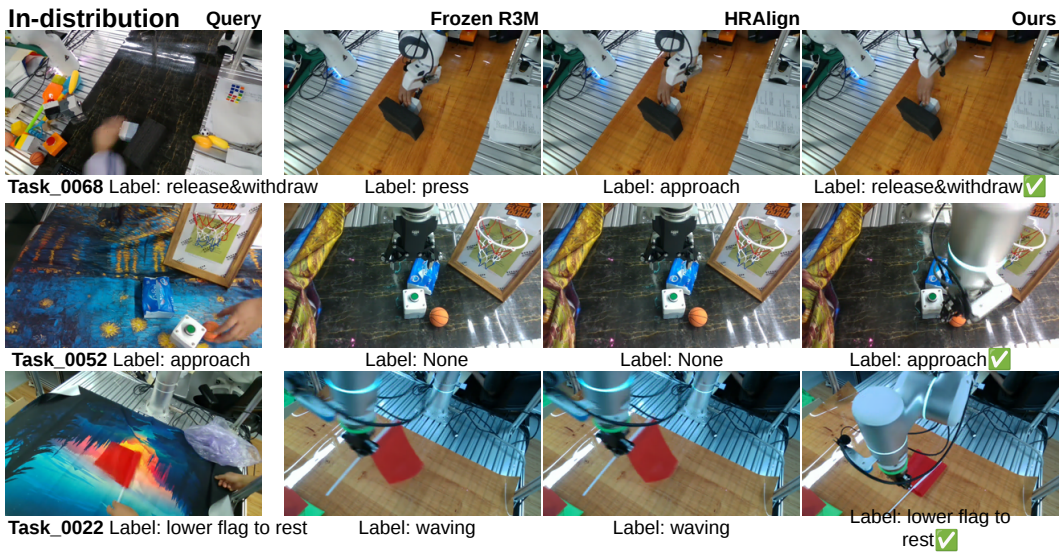


Figure 8: Human-to-robot task-progress retrieval examples from RH20T. Each row shows a human query and the nearest robot frames retrieved by Frozen R3M, HRAlign, and our method. Green checks indicate correct retrievals.

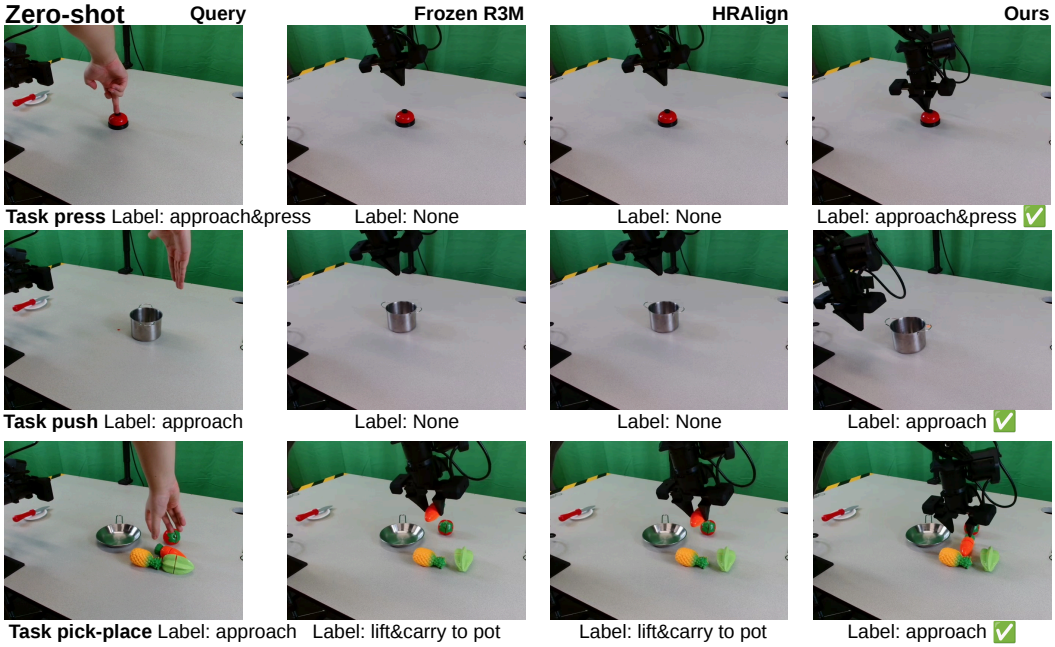


Figure 9: Human-to-robot task-progress retrieval examples from the self-collected zero-shot data. Each row shows a human query and the nearest robot frames retrieved by Frozen R3M, HRAlign, and our method. Green checks indicate correct retrievals. "None" denotes frames outside the annotated progress segments.

policy training but evaluates the trained policy on 23 task categories that are absent from the training set, as summarized in Table 12. Level-1 contains 13 unseen tasks that share partial object or action semantics with the training tasks, thereby evaluating transfer through recombinations of familiar semantics. Level-2 contains 10 tasks with previously unseen object and action semantics,

Table 10: Frame annotations for the ten selected RH20T human–robot pairs. Each task has one human-robot pair. Stage names are task-specific.

Task	Domain	Outside	Annotated	S1	S2	S3	S4
task_0010: S1: approach; S2: grasp and lift; S3: move handset away; S4: retreat							
task_0010	Human	0–41; 69–85	42–68	42–46	46–53	53–64	64–68
task_0010	Robot	0–519	520–938	520–600	600–680	680–880	880–938
task_0015: S1: approach; S2: lift and aim; S3: striking; S4: return stick and release							
task_0015	Human	0–43	44–69	44–47	47–55	55–64	64–69
task_0015	Robot	0–199	200–315	200–220	220–247	247–280	280–315
task_0016: S1: approach; S2: lift and aim cue; S3: stroke; S4: ball rolls to rest							
task_0016	Human	0–39	40–90	40–46	46–68	68–82	82–90
task_0016	Robot	0–119	120–379	120–132	132–244	244–334	334–379
task_0022: S1: approach; S2: grasp and lift; S3: waving; S4: lower flag to rest							
task_0022	Human	0–23	24–70	24–32	32–41	41–66	66–70
task_0022	Robot	0–129	130–347	130–147	147–197	197–334	334–347
task_0052: S1: approach; S2: grasp and lift; S3: carry and shoot							
task_0052	Human	0–26; 53–75	27–52	27–32	32–37	37–52	–
task_0052	Robot	0–49	50–280	50–86	86–188	188–280	–
task_0065: S1: approach and pick; S2: transport brick; S3: place brick							
task_0065	Human	0–46; 107–134	47–106	47–61	61–90	90–106	–
task_0065	Robot	0–139	140–338	140–170	170–255	255–338	–
task_0068: S1: approach; S2: press; S3: release and withdraw							
task_0068	Human	0–95; 124–128	96–123	96–101	101–116	116–123	–
task_0068	Robot	0–169	170–258	170–185	185–205	205–258	–
task_0092: S1: approach; S2: turn handle; S3: release and withdraw							
task_0092	Human	0–37; 69	38–68	38–49	49–59	59–68	–
task_0092	Robot	0–59	60–181	60–70	70–140	140–181	–
task_0105: S1: approach; S2: lift and carry to shelf; S3: release and retreat							
task_0105	Human	0–67; 192–211	68–191	68–102	102–172	172–191	–
task_0105	Robot	0–169	170–476	170–240	240–404	404–476	–
task_0222: S1: approach; S2: lift and carry lid; S3: seat lid and hold							
task_0222	Human	0–22	23–137	23–48	48–120	120–137	–
task_0222	Robot	0–249	250–861	250–300	300–746	746–861	–

requiring stronger cross-task zero-shot generalization. AGNOSTOS therefore evaluates whether a policy can transfer and recombine knowledge acquired from demonstrations of seen tasks to unseen tasks for which no downstream demonstrations are provided. We follow the downstream training and evaluation protocol of RVT and RVT-2 (Goyal et al., 2023; 2024).

B.4 Real World Environment

We conduct real-world experiments using a Trossen WidowX AI arm. The policy state is a seven-dimensional vector comprising six joint angles and a scalar gripper state. At a control frequency of 20 Hz, the policy predicts seven-dimensional absolute targets for the six robot joints and the gripper. These targets are executed using Trossen's original low-level controller and official drivers.

Visual observations are captured using a fixed external Logitech C920 camera positioned above and behind the robot. This configuration provides a third-person, approximately top-down view of the workspace. The camera records RGB images at a resolution of 640×480 and a frame rate of 20 Hz. We do not use depth observations, a wrist-mounted camera, or image augmentation.

As illustrated in Figure 10, we evaluate four tasks: *Pick Carrot into Pot*, *Press Button*, *Push Pot into Marked Region*, and *Open Pot by Lifting the Lid*. These tasks cover four manipulation skills: pick-and-place, button pressing, goal-directed pushing, and object opening (Yu et al., 2020; James et al., 2020). For each task, we collect 40 demonstrations through manual teleoperation, independently

Table 11: Frame annotations for the four self-collected zero-shot tasks, with five human and five robot videos per task. Stage names are task-specific.

Task	Demo	Domain	Outside	Annotated	S1	S2	S3
carrot: S1: approach; S2: lift and carry to pot; S3: release and retreat							
carrot	001	Human	0–78; 155–164; 169–203	79–154; 165–168	79–119	119–154	165–168
carrot	001	Robot	247–299	0–246	0–155	155–226	226–246
carrot	002	Human	0–86; 169–203	87–168	87–113	113–156	156–168
carrot	002	Robot	253–299	0–252	0–142	142–232	232–252
carrot	003	Human	0–81; 174–218	82–173	82–108	108–157	157–173
carrot	003	Robot	267–299	0–266	0–138	138–246	246–266
carrot	004	Human	0–60; 155–196	61–154	61–93	93–141	141–154
carrot	004	Robot	254–299	0–253	0–157	157–233	233–253
carrot	005	Human	0–50; 145–180	51–144	51–78	78–127	127–144
carrot	005	Robot	285–299	0–284	0–166	166–264	264–284
open: S1: approach; S2: lift lid and set aside; S3: retreat							
open	001	Human	0–60; 151–224	61–150	61–94	94–138	138–150
open	001	Robot	233–299	0–232	0–135	135–212	212–232
open	002	Human	0–55; 145–180	56–144	56–95	95–131	131–144
open	002	Robot	248–299	0–247	0–140	140–227	227–247
open	003	Human	0–55; 143–181	56–142	56–93	93–131	131–142
open	003	Robot	253–299	0–252	0–148	148–232	232–252
open	004	Human	0–59; 131–197	60–130	60–89	89–118	118–130
open	004	Robot	260–299	0–259	0–153	153–239	239–259
open	005	Human	0–39; 120–153	40–119	40–70	70–109	109–119
open	005	Robot	273–299	0–272	0–177	177–252	252–272
press: S1: approach and press; S2: release and retreat							
press	001	Human	0–43; 93–141	44–92	44–72	72–92	–
press	001	Robot	130–159	0–129	0–119	119–129	–
press	002	Human	0–51; 96–136	52–95	52–87	87–95	–
press	002	Robot	125–159	0–124	0–114	114–124	–
press	003	Human	0–39; 88–127	40–87	40–80	80–87	–
press	003	Robot	131–159	0–130	0–120	120–130	–
press	004	Human	0–56; 116–153	57–115	57–104	104–115	–
press	004	Robot	144–159	0–143	0–133	133–143	–
press	005	Human	0–77; 128–162	78–127	78–118	118–127	–
press	005	Robot	131–159	0–130	0–120	120–130	–
push: S1: approach; S2: push pot to area; S3: release and retreat							
push	001	Human	0–54; 114–190	55–113	55–80	80–104	104–113
push	001	Robot	195–239	0–194	0–132	132–174	174–194
push	002	Human	0–42; 102–141	43–101	43–66	66–91	91–101
push	002	Robot	194–239	0–193	0–119	119–173	173–193
push	003	Human	0–34; 90–123	35–89	35–59	59–81	81–89
push	003	Robot	213–239	0–212	0–138	138–192	192–212
push	004	Human	0–33; 87–121	34–86	34–53	53–76	76–86
push	004	Robot	195–239	0–194	0–143	143–174	174–194
push	005	Human	0–34; 80–113	35–79	35–51	51–70	70–79
push	005	Robot	205–239	0–204	0–124	124–184	184–204

randomizing the position of the task-relevant object for each demonstration. During evaluation, each policy is tested over 20 episodes per task, with the object position resampled for every episode.

Table 12: Seen training and unseen testing task categories in the AGNOSTOS benchmark. The 23 testing categories are **disjoint** from the 18 training categories.

Split	# Tasks	Task categories
Training	18	Close jar; insert onto square peg; light bulb in; meat off grill; open drawer; place cups; place shape in shape sorter; place wine at rack location; push buttons; put groceries in cupboard; put item in drawer; put money in safe; reach and drag; slide block to color target; stack blocks; stack cups; sweep to dustpan of size; and turn tap.
Testing Level-1	13	Put toilet roll on stand; put knife on chopping board; close fridge; close microwave; close laptop lid; phone on base; toilet seat down; lamp off; lamp on; put books on bookshelf; put umbrella in umbrella stand; open grill; and put rubbish in bin.
Testing Level-2	10	Take USB out of computer; take lid off saucepan; take plate off colored dish rack; basketball in hoop; scoop with spatula; straighten rope; turn oven on; beat the buzz; water plants; and unplug charger.

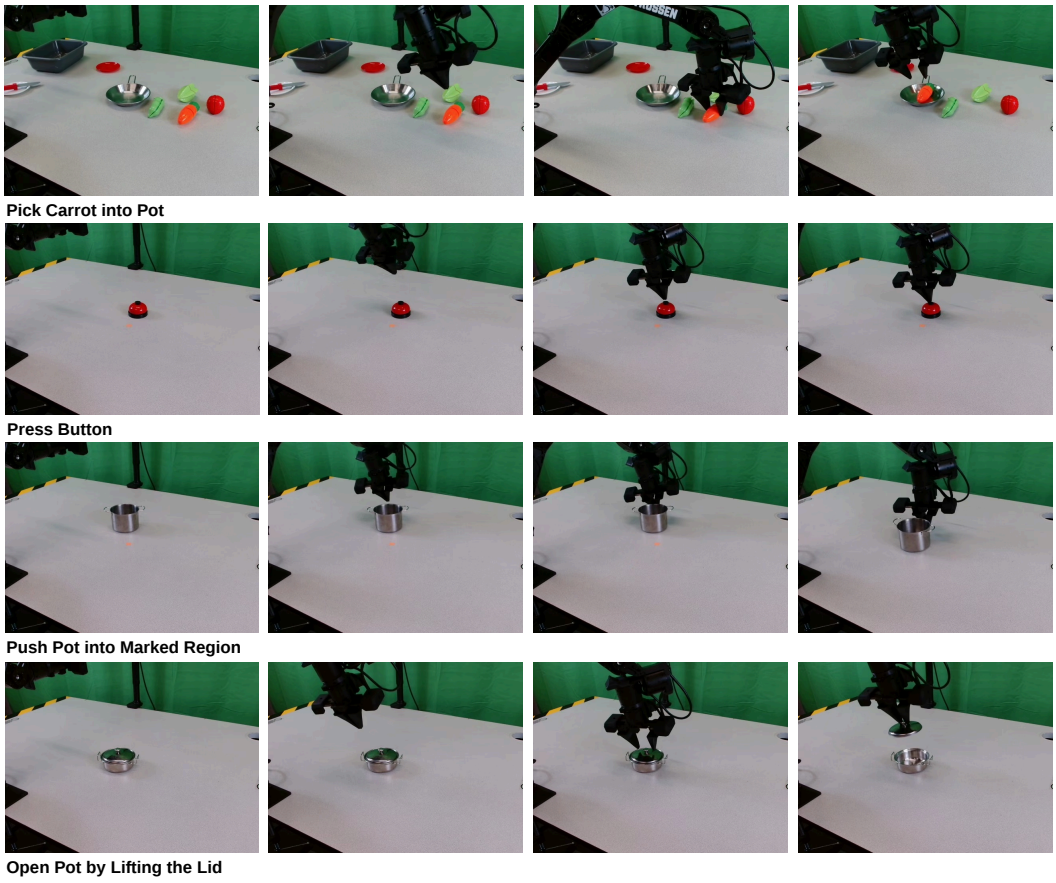


Figure 10: Real-world manipulation tasks evaluated using the Trossen WidowX AI arm.