HARP-VLA: Human-Robot Aligned Representation Learning for Vision-Language-Action Model

Xiang Zhu[1,2*] **Puzhen Yuan**[1*] **Yichen Liu**[1*] **Jianyu Chen**[1,2†]
[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, China
[2] Shanghai Qi Zhi Institute, China
{zhuxiang24, ypz21, liu-yc22}@mails.tsinghua.edu.cn
jianyuchen@mail.tsinghua.edu.cn
*Equal contribution, †Corresponding author.

Abstract: Learning generalizable vision-language-action (VLA) models from large-scale human videos is promising but challenging due to cross-embodiment discrepancies in both visual observations and executable actions. While latent action models reduce the action execution gap by learning action abstractions, they still rely on visual features. Thus, misaligned human and robot visual representations can lead to inconsistencies in policy inputs and induce domain-dependent latent actions, hindering effective co-training with human videos. To address this, we propose HARP, a human-robot aligned representation learning framework for more effective VLA pretraining from human videos. Specifically, HARP uses paired human-robot demonstrations as cross-embodiment bridges and abundant unpaired human and robot videos as a scalable dynamics supervision data source. It trains a robot-adapted visual encoder and a latent action model with manipulation-centric auxiliary cues and a source-relative pair-discriminative alignment loss, which adapts robot representations toward human semantics while preserving pair-level discrimination. The learned aligned vision encoder and latent action model provide a unified vision and action representation for VLA-style policy learning, where human and robot videos provide vision-language-to-latent-action supervision and a lightweight robot action head grounds latent actions into executable commands. Experiments on feature visualization, simulation, and real-world manipulation show improved human-robot alignment and downstream policy performance, achieving 4.481 average length on CALVIN ABC→D and a 7.1 percentage-point improvement in real-world success rate over the strongest evaluated baseline. Code and demo are available at https://github.com/anonymity35/HARP-VLA.

Keywords: Vision-Language-Action Model, Latent Action Model, Representation Learning, Human Videos

1 Introduction

Learning generalizable robot policies typically requires large-scale teleoperated demonstrations, which are expensive to collect and difficult to scale. In contrast, abundant human manipulation videos contain rich skill knowledge, offering a promising source of supervision for robot learning. However, directly leveraging human videos for robot learning presents two primary challenges. The action execution gap makes human motions difficult to translate into executable robot actions, while the visual representation gap causes similar manipulation dynamics to be encoded into separate human and robot feature manifolds, hindering effective co-training across the two domains, as shown in Fig. 1(up). As a result, policies may learn domain-specific representations rather than reusable skill abstractions, raising a central question: how can we learn a unified policy representation from human and robot videos under cross-domain discrepancies and weak supervision?

Recent advances in Latent Action Models (LAMs) [1, 2, 3, 4] mitigate the action execution gap by learning latent transition codes from temporally adjacent frames instead of embodiment-specific motor commands. However, because latent actions are grounded in visual observations, misaligned human and robot features can still make them domain-dependent, limiting human-robot co-training and downstream policy learning. Existing efforts address the visual representation gap through

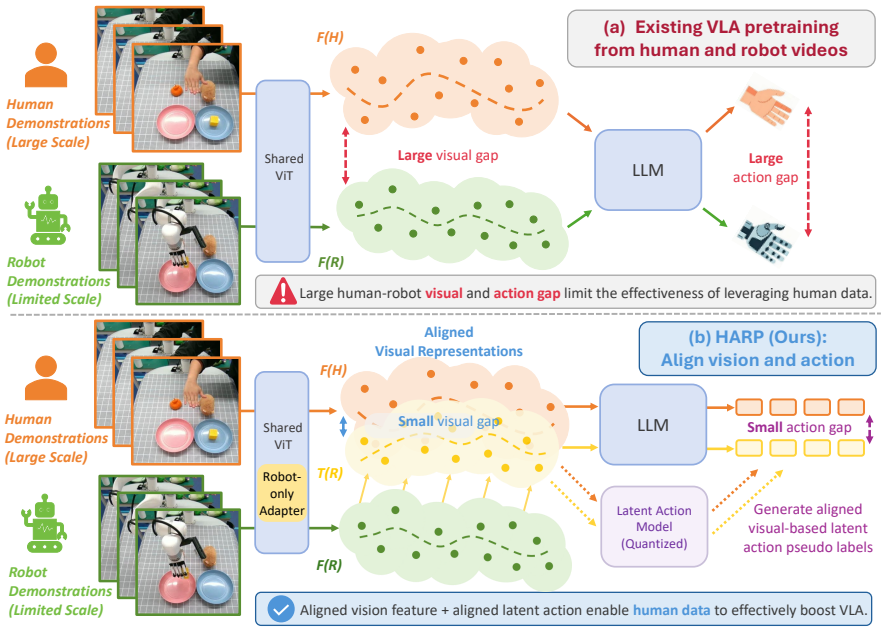

Figure 1: **Motivation and overview of HARP. Top:** Existing VLA pretraining encodes human and robot demonstrations into separated visual representations and suffers from a large action gap, limiting human data use. **Bottom:** HARP jointly aligns visual representations and latent actions using limited paired human-robot demonstrations and scalable unpaired videos, enabling human data to effectively improve VLA pretraining.

sparse annotations [5], image-level human-to-robot substitution [6, 7], or video-level representation adaptation [8]. However, they do not explicitly couple visual alignment with frame-level manipulation dynamics and latent action learning. Therefore, a general and scalable framework is still needed to jointly unify visual states and latent actions across human and robot videos, especially under limited paired supervision and abundant unpaired data.

To address this problem, we propose HARP, a human-robot aligned representation learning framework for robot policy learning from human videos. HARP uses limited paired human-robot demonstrations as cross-embodiment bridges and unpaired videos as scalable dynamics supervision. It anchors the human-side visual representation to the pretrained encoder, adapts the robot-side encoder with lightweight modules, and jointly learns latent actions over both paired and unpaired videos. Together with manipulation-centric auxiliary cues and a source-relative pair-discriminative alignment loss that preserves pair-level discrimination, HARP maps human and robot demonstrations into a unified visual space and latent action space, as shown in Fig. 1(down). By reducing cross-domain visual and action discrepancies, HARP provides a unified interface for policy learning from both human and robot videos, enabling more effective use of human video supervision. The learned aligned encoder and latent action model provide a unified interface for VLA pretraining. In this work, we instantiate this interface with a Prismatic/OpenVLA-OFT-style VLA backbone, where human and robot videos provide vision-language-to-latent-action supervision. A lightweight action head trained on real robot data then translates the learned latent actions into executable robot commands. The core contributions of this work are threefold:

1. We propose HARP, a three-stage framework that jointly learns human-robot aligned visual representations and latent actions for VLA pretraining from large-scale human videos.
2. We introduce a source-relative pair-discriminative alignment loss tailored to HARP, combined with manipulation-centric auxiliary cues, to align robot representations toward paired human semantics while maintaining pair-level discriminability.
3. We demonstrate improved visual and latent-action alignment, as well as consistent downstream gains, across representation analysis, simulated benchmarks, and real-world manipulation tasks.

2 Related Works

Vision-Language-Action Models for Robot Manipulation. Vision-Language-Action (VLA) models provide a unified framework for robot manipulation by modeling visual observations, language

instructions, and actions as Transformer-based token sequences. RT-1 [9] and RT-2 [10] show that large-scale robot demonstrations and web-scale vision-language pretraining improve cross-task generalization and semantic reasoning. Recent open-source VLAs further scale this paradigm toward multi-task, cross-embodiment, and continuous-action policy learning. OpenVLA [11] and Octo [12] leverage pretrained language and visual backbones, while RDT-1B [13], UniVLA [2], π_0 [14], and $\pi_{0.5}$ [15] extend VLA models to heterogeneous action spaces and continuous action generation.

Learning from Human Videos via Cross-Embodiment Alignment. Human videos provide demonstrations for robot learning, but are limited by embodiment gaps. Existing methods address cross-embodiment gaps through trajectory cues, object-motion modeling, or representation alignment. RT-Trajectory [16], MimicPlay [17], and Learning by Watching [18] use hand keypoint trajectories to guide policies. Object-centric methods such as Im2Flow2Act [19] and Dream2Flow [20] model object motion as an embodiment-invariant signal. More recent work reduces the human–robot domain gap through visual transformation or representation alignment. EgoMimic [5] uses visual alignment cues, RoVi-Aug [21] and DexUMI [7] generate embodiment-consistent observations, and HR-Align [8] learns shared feature spaces with contrastive objectives. These studies show that human videos can benefit robot manipulation when cross-embodiment variation is reduced through object or motion cues, visual observations, or learned representations.

Latent Action Models for Cross-Embodiment Policy Learning. Since human videos usually lack executable action labels, recent studies learn embodiment-agnostic latent actions from unlabeled or weakly labeled data. Rather than predicting robot commands, they learn shared motion embeddings as a unified action space for cross-embodiment transfer. LAPA [1] learns discrete latent action codes from large-scale unlabeled videos and maps them to robot controls with limited robot data, while UniVLA [2] improves this formulation by emphasizing task-relevant motion and suppressing irrelevant dynamics. UniSkill [3] and IGOR [4] further introduce inverse and forward dynamics objectives to learn motion-centric embeddings shared by humans and robots. Overall, latent action modeling provides a scalable interface between human videos and robot manipulation policies.

3 Methods

HARP is a three-stage framework for transferring human demonstration knowledge to robot policy learning through aligned visual representations and latent actions. Stage 1 jointly learns a robot-adapted visual encoder and a latent action model (LAM) from paired and unpaired videos; Stage 2 uses the learned encoder and LAM-generated latent-action labels to pretrain a VLA policy; and Stage 3 finetunes the policy on robot demonstrations with a lightweight real-action head.

3.1 Training Data and Preprocessing

Mixed paired and unpaired demonstrations. Our training data consist of paired and unpaired manipulation videos, $\mathcal{D} = \mathcal{D}_p \cup \mathcal{D}_u$. The paired set is defined as $\mathcal{D}_p = \{(H_i, R_i, l_i)\}_{i=1}^{N_p}$, where H_i and R_i are paired human and robot videos of the same task with instruction l_i. The unpaired set is $\mathcal{D}_u = \{(V_j, l_j)\}_{j=1}^{N_u}$, where V_j is a video from embodiment $e_{V_j} \in \{h, r\}$. Paired videos provide cross-embodiment supervision, while unpaired videos enrich task-relevant dynamics.

Shared task cues across embodiments. Human and robot demonstrations differ in appearance and morphology, but often share task-level cues such as instructions, object motion, and coarse agent motion. For each video X, we extract auxiliary annotations $A_X = \{K_X, E_X\}$, where K_X denotes 2D object position tracks, E_X denotes 2D human wrist or robot end-effector trajectory. Object tracks capture manipulation effects, while wrist/end-effector trajectories provide a coarse proxy for motion intent. These cues regularize the latent actions to capture motion dynamics rather than embodiment-specific appearance, especially for unpaired videos without cross-embodiment supervision.

Temporally aligned paired videos. For paired videos, although similar motion patterns are enforced during data collection, execution speed and subtask duration remain difficult to keep consistent. Thus, we perform dynamic time warping (DTW) over E_X to obtain temporal correspondence maps and sample matched human-robot transitions on the robot timeline.

Finally, processed $\mathcal{D}_p = \{(\hat{H}_i, \hat{R}_i, l_i, A_{\hat{H}_i}, A_{\hat{R}_i})\}_{i=1}^{N_p}$, where $\hat{\cdot}$ denotes temporally aligned, which is omitted hereafter for simplicity, and $\mathcal{D}_u = \{(V_j, l_j, A_{V_j})\}_{j=1}^{N_u}$. Paired videos support cross-embodiment prediction and alignment, while unpaired videos support self-prediction and auxiliary supervision. Details of annotation extraction and DTW alignment are provided in Appendix A.1.

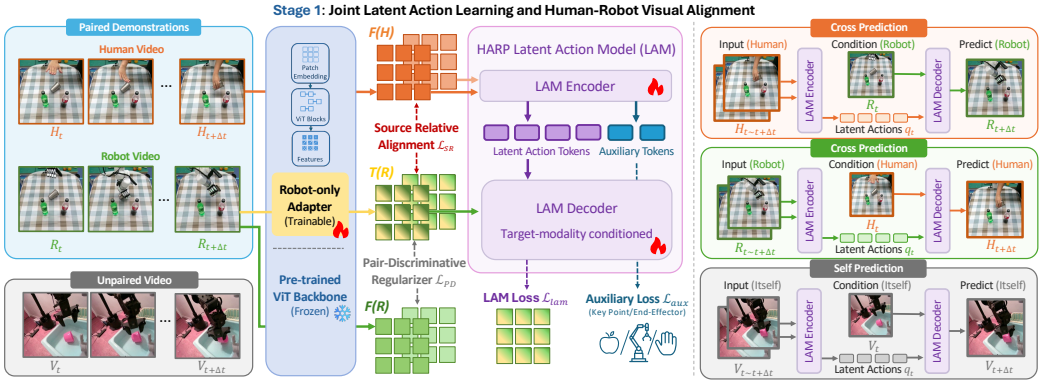

Figure 2: Stage 1: **Joint Visual and Latent-Action Alignment**. **Left:** Human to robot cross-prediction example. HARP jointly learns a robot-adapted visual encoder and a latent action model using paired videos for cross-prediction and alignment, and unpaired videos for self-prediction. Auxiliary cues guide latent-action learning. Source-Relative (SR) loss $\mathcal{L}_{\text{SR}}$ aligns robot features toward humans, while Pair-Discriminative (PD) regularizer $\mathcal{L}_{\text{PD}}$ preserves representation structure. **Right:** Examples of cross-prediction and self-prediction.

3.2 Joint Visual and Latent-Action Alignment

Embodiment-aware visual encoding. Given a video X with embodiment $e_X \in \{h, r\}$, we encode frames into patch-level visual tokens using embodiment-aware encoder composed of the frozen original visual encoder F and robot-adapted encoder T_θ (see Appendix A.3 for architecture),

$$\Phi_\theta(X, e_X) = \begin{cases} F(X), & e_X = h, \\ T_\theta(X), & e_X = r. \end{cases}$$

The resulting visual tokens and frozen teacher features are denoted by

$$Z_X = \Phi_\theta(X, e_X) = \{z_X^t\}_{t=1}^{T_X}, \qquad Z_{X0} = F(X),$$

where T_X is total frame number. This design anchors human videos to the pretrained visual space while adapting robot representations toward the same semantic space.

Latent-action prediction with Self- and Cross-Prediction. We instantiate the latent action module as a VQ-VAE-style inverse-and-forward dynamics model, where an encoder infers latent actions from visual transitions, and a decoder predicts future visual representations conditioned on the current observation and the quantized latent action. Each transition is represented by $N_q = 4$ discrete latent-action tokens quantized by a VQ codebook. Here, $t + \Delta t$ denotes the second frame in the sampled transition; details are provided in Appendix A.3.

Given a transition $(z_X^t, z_X^{t+\Delta t})$ and language instruction l_X, the encoder E_θ infers a continuous latent action, quantizes it with a codebook Q_θ, and the decoder D_θ predicts the target frame's patch-level visual tokens as the future representation, and $\tilde{z}_X^t$ denotes the decoder conditioning feature.

$$a_X^t = E_\theta(z_X^t, z_X^{t+\Delta t}, l_X), \qquad q_X^t = Q_\theta(a_X^t), \qquad \hat{Y}_X^t = D_\theta(\tilde{z}_X^t, q_X^t, l_X),$$

The decoder target Y_X^t and condition $\tilde{z}_X^t$ depends on whether the transition is sampled from paired or unpaired data. For an unpaired video V, HARP uses self-prediction: the model predicts the future representation within the same video $\tilde{z}_V^t = z_V^t, Y_V^t = z_V^{t+\Delta t}$. For paired videos (H, R), HARP performs cross-embodiment prediction. The latent action is inferred from the source transition, while the decoder is conditioned on the current representation of the target embodiment:

$$\tilde{z}_H^t = z_R^t, \quad Y_H^t = z_R^{t+\Delta t}; \qquad \tilde{z}_R^t = z_H^t, \quad Y_R^t = z_H^{t+\Delta t}.$$

This cross-prediction objective forces latent actions extracted from one embodiment to explain future dynamics of the other, thereby coupling latent-action learning with human-robot alignment.

Training objective. Stage-1 training combines three objectives:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{lam}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}.$$

The latent action prediction loss learns discrete latent actions through self- and cross-prediction, the auxiliary loss injects manipulation-centric shared cues, and the alignment loss consists of a source-relative term and a pair-discriminative regularizer to adapt robot representations toward human semantics while maintaining pair-level discrimination.

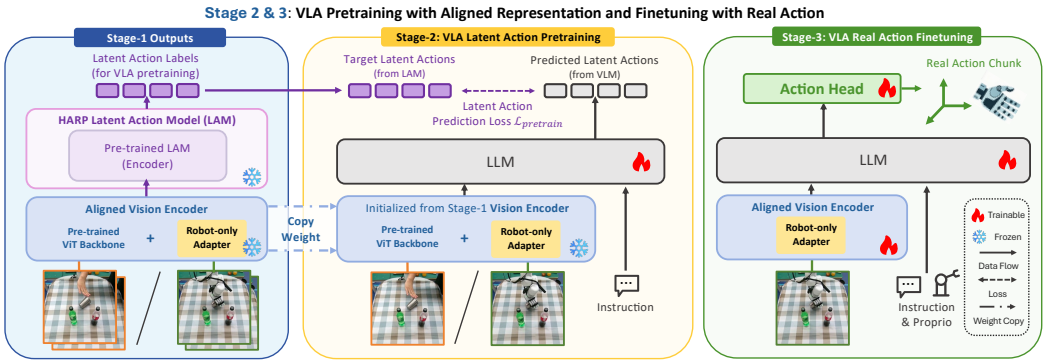

Figure 3: **Pretrain and finetune.** Stage-2: **Pretrain**. The Stage-1 LAM is used to produce human-robot aligned latent actions, and the aligned vision encoder is used to replace the VLM vision encoder. Stage-3: **Finetune**. A trainable action head is employed to convert latent action embeddings to executable real actions.

Latent-action prediction. Given the decoder prediction $\hat{Y}_X^t$ and target Y_X^t, we optimize

$$\mathcal{L}_{\text{lam}} = \mathbb{E}_{t<T_X, X\sim\mathcal{D}} \|\hat{Y}_X^t - Y_X^t\|_2^2 + \mathcal{L}_{\text{vq}},$$

where $\mathcal{L}_{\text{vq}}$ is the standard VQ codebook and commitment loss. This term trains the latent action to explain future visual dynamics under both self-prediction and cross-embodiment prediction.

Shared-cue auxiliary loss. To encourage task-relevant dynamics, we regularize latent-action learning with object-centric point tracks and wrist/end-effector trajectories. We introduce shared-cue auxiliary tokens in the latent-action encoder to predict these shared cues, and supervise them with

$$\mathcal{L}_{\text{aux}} = \lambda_K \mathcal{L}_K + \lambda_E \mathcal{L}_E.$$

Here, $\mathcal{L}_K$ and $\mathcal{L}_E$ are Huber losses over visible object keypoints and wrist/end-effector positions, respectively. These losses bias the latent actions toward object motion and coarse agent motion, which are more shared across embodiments than raw appearance.

Source-relative pair-discriminative alignment loss. For paired demonstrations, the adapted robot representation should satisfy two constraints: it should improve over the frozen robot representation with respect to the paired human video, and the paired correspondence should remain discriminative against non-matching pairs. For paired videos (H, R), let $f^H = \rho(Z_H), f^{R0} = \rho(Z_{R0}), f^R = \rho(Z_R)$, where $\rho(\cdot)$ denotes video-level pooling.

Using cosine distance $d(u, v) = 1 - \cos(u, v)$, the source-relative term requires the adapted robot feature to improve over the frozen robot feature with respect to the paired human feature:

$$\mathcal{L}_{\text{SR}} = \mathbb{E}_{(H,R)\sim\mathcal{D}_p} \left[m_s + d(f^R, f^H) - d(f^{R0}, f^H)\right]_+,$$

where m_s is a fixed triplet margin. This source-relative formulation avoids forcing all paired features to collapse to an absolute distance target.

To preserve pair-level discrimination, we further add a pair-discriminative term

$$\mathcal{L}_{\text{PD}} = \mathbb{E}_{(H,R)\sim\mathcal{D}_p} \sum_{\alpha\in\{\text{R}\to\text{H},\text{H}\to\text{R}\}} \lambda_\alpha \left[m_t + d(f^R, f^H) - \bar{d}^\alpha\right]_+,$$

where $\bar{d}^{\text{R}\to\text{H}} = \mathbb{E}_{H'\neq H} d(f^R, f^{H'})$, $\bar{d}^{\text{H}\to\text{R}} = \mathbb{E}_{R'\neq R} d(f^{R'}, f^H)$, and m_t is a fixed margin.

The final alignment loss is

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{SR}} + \mathcal{L}_{\text{PD}}.$$

Detailed formulations of the training objectives are provided in Appendix A.2. We refer to the latent action model trained as HARP-LAM, and the aligned vision encoder as HARP-VE.

3.3 VLA Pretraining with Aligned Representations

Leveraging the latent action model and aligned vision encoder trained in stage-1, we can label video frame $x^t \in X$ with unified latent action $q_X^t = Q_\theta(E_\theta(z_X^t, z_X^{t+\Delta t}, l_X))$, where $z_X^t = \Phi_\theta(x^t, e_X)$, $z_X^{t+\Delta t} = \Phi_\theta(x^{t+\Delta t}, e_X)$, which can be employed to train a generalist policy, as shown in Fig 3.

We use latent action labels to pretrain a VLM into a VLA, which uses the same vision encoder as HARP-LAM. The architecture details can be found in the Appendix A.3. Specifically, the model π_θ receives language instruction l_X and visual input x^t, and outputs latent action tokens $\{\hat{q}_i\}_{i=1}^{N_q}$, and is optimized to minimize the cross-entropy loss

$$\mathcal{L}_{pretrain} = -\mathbb{E}_{(x^t, l_X)\sim\mathcal{D}} \left[\sum_{i=1}^{N_q} \log \pi_\theta(\hat{q}_i = q_{X,i}^t | x^t, l_X)\right],$$

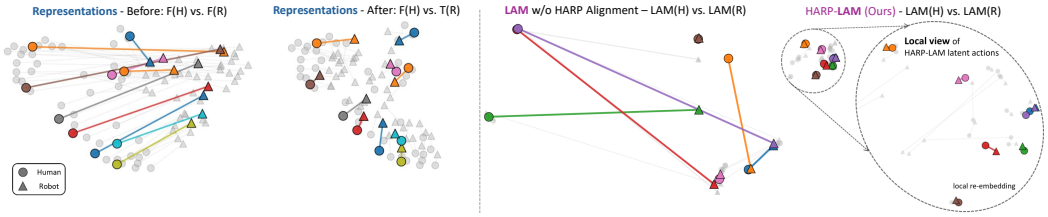

Figure 4: UMAP visualization of human-robot alignment. Left: visual representations before adaptation, $F(H)$ vs. $F(R)$, and after HARP adaptation, $F(H)$ vs. $T(R)$. Right: latent-actions without and with HARP-LAM alignment. Circles and triangles denote human and robot; colored pairs are highlighted for readability.

where N_q represents the number of latent action tokens. This approach leverages the original visual reasoning and language understanding capabilities of the pretrained VLM, while additionally equipping it with action modeling ability by training it to predict latent action indices.

In addition to directly pretraining on original VLM, we further propose copying the adapter weights learned in the stage-1 (3.2) to the vision encoder of VLM, as shown in Fig 3. In this way, we enable the vision encoder to achieve human–robot visual alignment as well as latent action alignment, facilitating effective pre-training on large-scale human videos for robot manipulation.

3.4 VLA Finetuning with Real Action

As the VLA is pretrained with latent actions, it cannot directly output real actions for control. To address this, we finetune it with a small amount of real-action robot data for downstream deployment. Specifically, we employ an action head that maps from the latent action embedding to normalized real action. We train the action head from scratch using L1 loss, and simultaneously use LoRA [22] to finetune the VLA backbone to achieve effective adaptation. We broadly refer to the policy model that has gone through these stages as HARP-VLA, with architecture details in Appendix A.3.

4 Experiments

In stage-1, we train HARP-LAM on human datasets OpenEgo [23], robot datasets Bridge-V2 [24], and human-robot paired datasets RH20T [25], Human2Robot [26] and a self-collected paired dataset on Robotera Xhand. In stage-2, we pretrain our VLA on the same datasets, using only video data and latent action labels extracted from HARP-LAM. In the last stage, we further fine-tune the HARP-VLA according to downstream tasks: CALVIN [27], and real-world experiment.

The goal of our experiments is to validate the effectiveness of the proposed HARP framework and test HARP-VLA's general control ability. We evaluate HARP from three perspectives: (1) whether it aligns human and robot visual representations and latent actions; (2) whether the aligned visual encoder improves downstream robot policy learning; and (3) whether the resulting HARP-VLA policy improves generalist manipulation performance in simulation and real-world settings.

4.1 Human-Robot Representation and Latent-Action Alignment

We first evaluate whether HARP reduces the human-robot representation gap before turning to downstream policy learning. We use complementary qualitative and quantitative evaluations, including UMAP visualization, paired human-robot cosine distance, cross-embodiment visual retrieval, and latent-action correspondence.

Baselines. We compare against the unadapted vision encoder and HR-Align-based [8] baselines. HR-Align (denoted as HR) uses its original task-aware pooling design, where the language instruction attends to video features before contrastive alignment. HR-Style removes task-aware pooling and applies the same contrastive loss directly on pooled features. We also evaluate HARP variants by changing alignment objective and pooling space. HARP-HR uses original HR-Align objective with task-aware pooling, while HARP-HR-Style applies HR-Align contrastive loss directly on pooled HARP features. HARP-L2 uses a naive L2 loss as the alignment loss on paired features. HARP-SR uses only the source-relative term, HARP-SRPD further adds the pair-discriminative term.

Comparison protocol. All HARP variants use the same backbone, robot-only adapter, data split and preprocess, and training budget; only the alignment objective or pooling space differs. HR-Align baselines use the same fused backbone and adapter but follow their original paired-data alignment setup. All representation metrics are evaluated on the same held-out paired videos with identical sampling, pooling, and cosine similarity. See Appendix A.4 for details.

Method	H2R R@1 ↑	R2H R@1 ↑	Avg. R@1 ↑
Unadapted	44.09	43.01	43.55
HR	45.16	45.16	45.16
HARP-HR	46.24	60.22	53.23
HARP-L2	70.97	52.69	61.83
HARP-SR	84.95	64.52	74.74
HARP-SRPD	**87.10**	**69.89**	**78.50**

Table 1: Bidirectional cross-embodiment retrieval on held-out paired demonstrations. H2R uses human videos as queries to retrieve paired robot videos, while R2H retrieves human from robot.

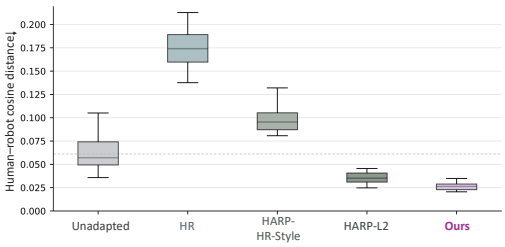

Figure 5: Paired human-robot cosine distance.

Qualitative visualization. Figure 4 provides a qualitative comparison of the learned visual representation and latent-action spaces. For visual representations, we compare the original feature space before adaptation, i.e., $F(H)$ versus $F(R)$, with the adapted feature space after HARP alignment, i.e., $F(H)$ versus $T(R)$. Before adaptation, human and robot demonstrations form visibly separated clusters, and even semantically matched pairs often remain far apart. After adaptation, paired human-robot samples become substantially closer, with a noticeably reduced cross-embodiment gap. A similar trend is observed in the latent-action space. Without HARP alignment, latent actions extracted from human and robot videos are dispersed and often poorly matched across embodiments. In contrast, HARP-LAM produces a more compact and better aligned latent-action space, where paired human and robot motions are mapped to nearby regions. These qualitative results suggest that HARP improves alignment at both the visual representation level and the latent-action level.

Paired human-robot cosine distance. We further measure geometric alignment using paired cosine distance. For unadapted encoder, we measure $d^{\text{orig}} = 1 - \cos(\rho(F(H)), \rho(F(R)))$. For adapted methods, we keep the human branch frozen and only adapt the robot branch, measuring $d^{\text{adapt}} = 1 - \cos(\rho(F(H)), \rho(T_\theta(R)))$. As shown in Fig. 5, HR-Align-based objectives do not reduce the paired distance when evaluated in the same mean-pooled visual representation space. In contrast, HARP variants substantially reduce the paired distance in the evaluation space, indicating that the learned robot-adapted features are geometrically closer to their paired human counterparts.

Cross-embodiment retrieval. Paired distance measures absolute geometric alignment, but it does not evaluate whether the learned space remains discriminative across different demonstrations. We therefore perform cross-embodiment retrieval on held-out paired videos. Given a human query, H2R retrieval ranks all robot videos by cosine similarity and checks whether the paired robot is retrieved; R2H retrieval is defined symmetrically. We report Recall@1. As shown in Table 1, HARP-SRPD improves from 43.55 to 78.50 in averaged R@1 over the unadapted encoder and outperforms HR-Align by a large margin. Compared with HARP-SR, adding the pair discriminative term further improves retrieval, indicating stronger cross-embodiment discriminability.

Variant	**Pair Pred.**	**Unpair.**	**Aux**	**Vis. R@1↑**	**Lat. R@1↑**	**CA-gap↑**
Full HARP	S+C	Yes	Yes	**78.50**	**36.29**	**30.67**
Self-only	S	Yes	Yes	73.93	26.62	22.24
Cross-only	C	Yes	Yes	75.10	28.03	25.16
Paired-only	S+C	No	Yes	75.16	32.06	30.04
w/o Aux	S+C	Yes	No	73.37	34.07	26.71

Table 2: Controlled Stage-1 component ablations. S/C denote self-/cross-prediction on paired samples; unpaired samples use self-prediction.

Cross-embodiment latent-action correspondence. We further evaluate alignment in the latent-action space by applying the same bidirectional retrieval protocol to transition-level latent-action embeddings, reporting the averaged H2R/R2H Recall@1 as Lat. R@1. We additionally report the code-agreement gap (CA-gap), which measures how often temporally corresponding human-robot transitions share discrete VQ codes relative to shuffled pairs. As shown in Table 2, the latent retrieval and code-agreement results provide quantitative evidence of consistent cross-embodiment latent-action correspondence.

4.2 Component Ablations and Paired-Data Scaling

Component ablations. As shown in Table 2, combining self- and cross-prediction consistently improves both visual and latent-action correspondence over either objective alone. Removing unpaired

Paired Data	Vis. R@1↑	Lat. R@1↑	RLBench↑	CALVIN↑
100% (∼24M)	**78.50**	**36.29**	**46.59**	**4.481**
25% (∼6M)	77.36	28.43	42.10	4.354
5% (∼1.2M)	72.95	24.40	40.70	4.214

Table 3: Paired-data scaling. Parentheses denote processed paired frames; unpaired data and optimization budget are fixed.

videos or auxiliary supervision also degrades alignment, indicating that these components provide complementary supervision.

Paired-data scaling. We further examine HARP under substantially reduced paired supervision while keeping unpaired data and the optimization budget fixed. As shown in Table 3, HARP retains strong visual and latent-action alignment as well as downstream policy performance even with only 5% of the paired data, demonstrating graceful scaling with the amount of paired supervision. Additional experimental details are provided in Appendix A.5.

Method	Avg. ↑
Unadapted	37.56
HR	39.70
HR-Style	38.22
HARP-HR	35.11
HARP-HR-Style	40.07
HARP-L2	40.78
HARP-SR	43.41
HARP-SRPD	**46.59**

Table 4: Average success rate (%) on 18 RLBench tasks using frozen visual encoders.

Model	Pick ↑	Push ↑	Press ↑	Flip ↑	Avg. ↑
π_0 [14]	58.3	75.0	56.7	35.0	56.3
$\pi_{0.5}$ [15]	71.7	**83.3**	68.3	53.3	69.2
OpenVLA [11]	0.0	23.3	18.3	0.0	10.4
UniVLA [2]	38.3	61.7	31.7	21.7	38.4
OpenVLA-OFT [28]	51.7	71.7	76.7	43.3	60.9
HARP-VLA (L2)	70.0	71.7	81.7	56.7	70.0
HARP-VLA (w/o F.)	**76.7**	80.0	78.3	58.3	73.3
HARP-VLA (Ours)	**76.7**	81.7	**85.0**	**61.7**	**76.3**

Table 5: Real-world manipulation success rate (%) averaged over 60 trials per task. L2 indicates using L2 alignment loss instead of SRPD. F. indicates freezing vision encoder during VLA pretraining.

4.3 Robot Policy Learning with Aligned Visual Representations

We next evaluate whether the aligned visual representation is useful for downstream robot policy learning. To evaluate the ability of HARP-VE for downstream manipulation policy learning, we follow HR-Align [8] to evaluate different methods on RLBench [29], which is challenging for its multi-task setting with significant task diversity and difficulty. All methods use the same policy architecture, training data, action space, and optimization budget. Thus, differences in success rate mainly reflect the quality of the learned visual representation.

As shown in Table 4, HARP-SRPD achieves the best policy performance, improving the average success rate from 37.56 to 46.59 over the unadapted encoder. Compared with HARP-SR, adding the pair-discriminative term further improves downstream success, suggesting that stronger cross-embodiment discriminability translates into more useful robot features.

4.4 HARP-VLA Policy Evaluation

Experiment Setup. We pretrain our HARP-VLA to predict aligned latent actions, and finetune it to execute tasks CALVIN [27] and real-world. Specifically, we choose the most challenging CALVIN ABC→D settings in simulation, and finetune it to control an Xarm7 with Robotera Xhand in real-world. Across all experimental settings, HARP-VLA takes third-view and wrist camera images, task instructions, and proprioceptive states as input, and outputs 10-step real action chunk. Further simulation and real-world details are in Appendix A.7.

Baselines. We compare HARP-VLA against 5 representative baselines, including OpenVLA, UniVLA, OpenVLA-OFT, π_0, and $\pi_{0.5}$, where UniVLA is similar to our training method, and OpenVLA-OFT is similar to our architecture. We also evaluate HARP with L2 alignment loss and without vision encoder freezing in pretraining as ablations. Baseline details are in Appendix A.7.

Due to differences in backbone, pretraining data, and visual input configurations, these results should be interpreted as system-level comparisons rather than fully controlled architectural comparisons.

Experiment Results. Tab. 5 and Tab. 6 show that HARP-VLA achieves an average length of 4.481 on CALVIN ABC→D and a mean success rate of 76.3% on the real-world tasks.

Table 6: Success rate (%) and average length on CALVIN benchmark. L2 indicates using L2 alignment loss instead of SRPD. F. indicates freezing the vision encoder in VLA pretraining.

Model	Task1 ↑	Task2 ↑	Task3 ↑	Task4 ↑	Task5 ↑	Avg. Len.↑
π_0 [14]	92.3	82.4	72.1	62.2	53.7	3.627
$\pi_{0.5}$ [15]	94.4	86.0	76.4	69.7	61.0	3.875
OpenVLA [11]	91.3	77.8	62.0	52.1	43.5	3.270
UniVLA [2]	95.4	85.5	75.4	66.9	56.5	3.800
OpenVLA-OFT [28]	94.2	86.4	78.0	70.4	62.7	3.917
HARP-VLA (L2)	95.8	89.7	81.3	72.8	64.8	4.044
HARP-VLA (w/o F.)	98.8	93.9	86.1	77.7	68.5	4.250
HARP-VLA (Ours)	**99.8**	**96.7**	**91.3**	**84.4**	**75.9**	**4.481**

In ablations, we compare SRPD with L2 alignment loss and study vision encoder freezing during pretraining, as shown in Tab. 5 and Tab. 6. Although the simple L2 alignment loss achieves competitive feature alignment and RLBench performance, SRPD performs better in VLA pretraining, improving performance by 0.437 on CALVIN average length and 6.3 percentage points in real-world success rate. Freezing the HARP-initialized vision encoder during Stage-2 pretraining further improves CALVIN average length from 4.250 to 4.481 and real-world success rate from 73.3% to 76.3%. This suggests that, after the robot-only adapter bridges the human-robot visual gap, freezing the vision encoder during latent-action pretraining helps preserve both the learned human-robot alignment and the web-scale pretrained visual features for robot manipulation.

5 Conclusions

We presented HARP-VLA, a framework which aligns human-robot visual representations and latent actions to improve VLA pretraining from human videos. By combining paired videos as cross-embodiment bridges, unpaired videos as dynamics supervision, and a source-relative pair-discriminative loss that aligns robot features while preserving pair-level separability, HARP learns an embodiment-aware visual encoder and latent-action model for downstream VLA training. Experiments on representation alignment, simulation benchmarks, and real-world manipulation show that paired bridge demonstrations, together with unpaired video dynamics supervision, provide an effective route toward VLA pretraining from human videos.

6 Limitations

HARP-VLA relies on task-paired human-robot demonstrations and auxiliary cues as cross-embodiment bridges. Although our scaling study shows strong performance under substantially reduced paired supervision, the absolute cost of collecting paired demonstrations remains non-negligible. The trajectory-based alignment currently assumes approximately compatible viewpoints and may degrade under severe tracking errors or large viewpoint/FoV shifts. Our current real-world evaluation is further limited to tabletop manipulation on a single robot platform. Future work will scale to larger and more diverse human video collections, improve robustness to noisy auxiliary cues, and validate HARP-VLA across more embodiments, longer-horizon tasks and bimanual manipulation.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback, which helped improve the clarity and experimental evaluation of this work. We also thank Robotera for providing access to the robotic hardware used in our real-world experiments.

References

[1] S. Ye, J. Jang, B. Jeon, S. J. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Lidén, K. Lee, J. Gao, L. S. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos. *ArXiv*, abs/2410.11758, 2024. URL https://api.semanticscholar.org/CorpusID:273351190.

[2] Q. Bu, Y. Yang, J. Cai, S. Gao, G. Ren, M. Yao, P. Luo, and H. Li. Univla: Learning to act anywhere with task-centric latent actions. *ArXiv*, abs/2505.06111, 2025. URL https://api.semanticscholar.org/CorpusID:278481174.

[3] H. Kim, J. Kang, H. Kang, M. Cho, S. J. Kim, and Y. Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. *ArXiv*, abs/2505.08787, 2025. URL https://api.semanticscholar.org/CorpusID:278535353.

[4] X. Chen, J. Guo, T. He, C. Zhang, P. Zhang, D. Yang, L. Zhao, and J. Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. 2024. URL https://api.semanticscholar.org/CorpusID:273811367.

[5] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233, 2024. URL https://api.semanticscholar.org/CorpusID:273707799.

[6] H. Li, I. Zhang, R. Ouyang, X. Wang, Z. Zhu, Z. Yang, Z. Zhang, B. Wang, C. Ni, W. Qin, et al. Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. *arXiv preprint arXiv:2509.22199*, 2025.

[7] M. Xu, H. J. Zhang, Y. Hou, Z. Xu, L. J. Fan, M. Veloso, and S. Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *ArXiv*, abs/2505.21864, 2025. URL https://api.semanticscholar.org/CorpusID:278960092.

[8] J. Zhou, T. Ma, K.-Y. Lin, R. Qiu, Z. Wang, and J. Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22551–22561, 2024. URL https://api.semanticscholar.org/CorpusID:270619804.

[9] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. A. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022. URL https://api.semanticscholar.org/CorpusID:254591260.

[10] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, K. Choromanski, T. Ding, D. Driess, K. A. Dubey, C. Finn, P. R. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, I. Leal, S. Levine, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. S. Ryoo, G. Salazar, P. R. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. H. Vuong, A. Wahid, S. Welker, P. Wohlhart, T. Xiao, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *ArXiv*, abs/2307.15818, 2023. URL https://api.semanticscholar.org/CorpusID:260293142.

[11] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, G. Lam, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *ArXiv*, abs/2406.09246, 2024. URL https://api.semanticscholar.org/CorpusID:270440391.

[12] O. M. Team, D. Ghosh, H. R. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, P. R. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. *ArXiv*, abs/2405.12213, 2024. URL https://api.semanticscholar.org/CorpusID:266379116.

[13] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *ArXiv*, abs/2410.07864, 2024. URL https://api.semanticscholar.org/CorpusID:273233386.

[14] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[15] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[16] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, P. Sundaresan, P. Xu, H. Su, K. Hausman, C. Finn, Q. H. Vuong, and T. Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *ArXiv*, abs/2311.01977, 2023. URL https://api.semanticscholar.org/CorpusID:265018996.

[17] C. Wang, L. J. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:257205825.

[18] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg. Learning by watching: Physical imitation of manipulation skills from human videos. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834, 2021. URL https://api.semanticscholar.org/CorpusID:231632575.

[19] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. *ArXiv*, abs/2407.15208, 2024. URL https://api.semanticscholar.org/CorpusID:271328597.

[20] K. Dharmarajan, W. Huang, J. Wu, L. Fei-Fei, and R. Zhang. Dream2flow: Bridging video generation and open-world manipulation with 3d object flow. *arXiv preprint arXiv:2512.24766*, 2025.

[21] L. Y. Chen, C. Xu, K. Dharmarajan, M. Z. Irshad, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *ArXiv*, abs/2409.03403, 2024. URL https://api.semanticscholar.org/CorpusID:272423529.

[22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[23] A. Jawaid and Y. Xiang. Openego: A large-scale multimodal egocentric dataset for dexterous manipulation. *arXiv preprint arXiv:2509.05513*, 2025.

[24] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[25] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[26] S. Xie, H. Cao, Z. Weng, Z. Xing, S. Shen, J. Leng, X. Qiu, Y. Fu, Z. Wu, and Y.-G. Jiang. Human2robot: Learning robot actions from paired human-robot videos. *ArXiv*, abs/2502.16587, 2025. URL https://api.semanticscholar.org/CorpusID:276575296.

[27] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

[28] M. J. Kim, C. Finn, and P. Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[29] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[30] X. Zhu, Y. Liu, H. Li, and J. Chen. Learning generalizable robot policy with human demonstration video as a prompt, 2025. URL https://arxiv.org/abs/2505.20795.

[31] Q. Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[32] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[33] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.

[34] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254, 2025.

[35] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

[36] M. Yadav and M. A. Alam. Dynamic time warping (dtw) algorithm in speech: a review. *International Journal of Research in Electronics and Computer Engineering*, 6(1):524–528, 2018.

[37] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

[38] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[40] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

A Appendix

A.1 Data Process

The objective of paired training data curation is to establish frame-level correspondence between human demonstration data and robot motion data, in order to provide frame-level visual clues for the following cross-prediction procedure. We propose to extract different types of keypoints as extra supervision and perform temporal alignment to better synchronize motion patterns for cross-prediction. As shown in Fig A1.

Keypoint extraction. For all human and robot videos, we extract two kinds of keypoints for supervision: object keypoint following object-flow-based methods [19, 20] and hand keypoint following bridge-domain-gap methods [5, 30].

Regarding object keypoint extraction, our pipeline is explicitly designed to handle severe occlusion, which is largely underexplored in prior work. If no language instruction is available, we first use Qwen3-VL-8B-Instruct [31] to generate a description of the manipulated object, then apply GroundingDINO [32] to localize its bounding box at the first occurrence, extract keypoints from this initial frame, and track them with TAPIR [33]. Although this design appears complex, it is necessary because existing pipelines typically fail under full occlusion and rarely address it explicitly. TAPIR is particularly suitable as it leverages historical context to infer current positions and provides occlusion and visibility scores, which enable us to attenuate unreliable predictions during occluded frames and thus improve overall robustness.

Towards hand keypoint extraction in human videos, we employ WiLoR [34], a monocular 3D hand reconstruction framework that regresses MANO [35] parameters from RGB input, from which we directly obtain the wrist's 2D coordinates on the image plane. As for hand keypoint extraction in robot videos, we leverage the camera extrinsics together with the recorded wrist pose in 3D space to project it onto the image plane, thereby computing the corresponding 2D coordinates in each frame through standard perspective projection.

Temporal alignment. For human–robot paired videos, although similar motion patterns are enforced during data collection, the execution speed and duration of subtask actions remain difficult to keep strictly consistent.

To address this temporal discrepancy, we use the Euclidean distance between the previously extracted hand keypoints as a similarity metric to characterize the correspondence between human and robot videos. We then apply Dynamic Time Warping (DTW) [36] to achieve strict frame-wise temporal alignment. Specifically, we take the more uniformly executed robot video as the temporal reference and resample the human video by filtering or duplicating frames according to the optimal DTW matching path. In this way, we obtain fully synchronized human-robot paired videos in terms of both duration and execution speed, reducing the need for precise temporal alignment during human paired-data collection and allowing the data acquisition process to focus primarily on matching motion patterns.

Training data summary. Table A1 summarizes the data and supervision used in each training stage. Scale denotes the number of frames used after preprocessing and filtering. For RH20T, due to heterogeneous camera viewpoints and variable pair quality, we use a selected subset rather than the full raw paired-video pool. For Ours-Real, only video observations are used in Stage 1/2, while real action labels are used only in Stage 3. Evaluation trials are held out from finetuning demonstrations.

A.2 Loss Details

This section provides the detailed formulations of the losses used in Stage 1. We denote a sampled transition from video X as $(z_X^t, z_X^{t+\Delta t})$, with language instruction l_X, where $t + \Delta t$ denotes the second frame in the sampled transition. The latent-action encoder produces continuous latent-action vectors a_X^t, which are quantized into q_X^t. The decoder prediction is denoted as $\hat{Y}_X^t$, and the corresponding target is Y_X^t, as defined in Sec. 3.2.

Given a transition, the decoder predicts

$$a_X^t = E_\theta(z_X^t, z_X^{t+\Delta t}, l_X), \qquad q_X^t = Q_\theta(a_X^t), \qquad \hat{Y}_X^t = D_\theta(\tilde{z}_X^t, q_X^t, l_X),$$

where $\tilde{z}_X^t$ is the decoder conditioning feature. For an unpaired video V, HARP uses self-prediction:

$$\tilde{z}_V^t = z_V^t, \qquad Y_V^t = z_V^{t+\Delta t}.$$

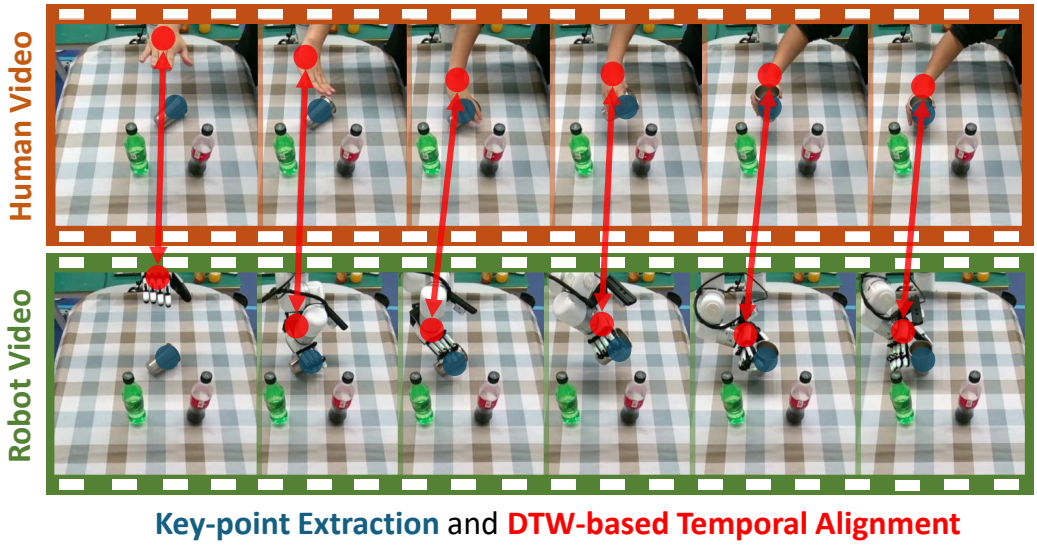

Figure A1: Paired data curation pipeline. We extract object-keypoint (blue points) for extra supervision, wrist position (red points) for supervision, and L2-distance-based dynamic time warping (red arrows).

Table A1: Dataset-level data usage in HARP-VLA. S1 denotes HARP-LAM/HARP-VE training, S2 denotes VLA latent-action pretraining, and S3 denotes downstream real-action finetuning.

Dataset	Embodiment	Data type	Scale used	S1	S2	S3
HOI4D	Human	Unpaired video	8.9M frames	Yes	Yes	No
OpenEgo	Human	Unpaired video	36.4M frames	Yes	Yes	No
Bridge-V2	Robot	Unpaired video	8.6M frames	Yes	Yes	No
Ours-Unpaired	DexHand	Unpaired video	3.6M frames	Yes	Yes	No
RH20T	Human-Robot	Paired video	8.16M frames	Yes	Yes	No
Human2Robot	Human-Robot	Paired video	9.9M frames	Yes	Yes	No
Ours-Paired	Human-DexHand	Paired video	5.7M frames	Yes	Yes	No
CALVIN	Robot / Sim.	Real-action demos	1.1M frames	No	No	Yes
Ours-Real	DexHand / Real	Real-action demos	0.5M frames	Yes	Yes	Yes

For a paired sample (H_i, R_i), HARP uses cross-embodiment prediction:

$$z_{H_i}^t = z_{R_i}^t, \quad Y_{H_i}^t = z_{R_i}^{t+\Delta t}; \qquad z_{R_i}^t = z_{H_i}^t, \quad Y_{R_i}^t = z_{H_i}^{t+\Delta t}.$$

Latent action prediction loss. The latent action prediction loss contains a future-representation prediction term and a standard VQ loss. For a transition from video X, we define

$$\ell_{\text{lam}}(X,t) = \left\| \hat{Y}_X^t - Y_X^t \right\|_2^2 + \ell_{\text{vq}}(X,t).$$

The VQ loss is

$$\ell_{\text{vq}}(X,t) = \left\| \text{sg}[a_X^t] - q_X^t \right\|_2^2 + \beta \left\| a_X^t - \text{sg}[q_X^t] \right\|_2^2,$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and β is the commitment weight.
Given a set of sampled transitions $\mathcal{B}$, the batch-level latent action prediction loss is

$$\mathcal{L}_{\text{lam}} = \frac{1}{|\mathcal{B}|} \sum_{(X,t)\in\mathcal{B}} \ell_{\text{lam}}(X,t).$$

Masked shared-cue auxiliary loss. To inject manipulation-centric cues, we introduce dedicated auxiliary tokens in the latent-action encoder. Let $u_{X,K}^\tau$ and $u_{X,E}^\tau$ denote the auxiliary token embeddings for object-centric keypoints and wrist/end-effector prediction at frame τ. Two lightweight prediction heads produce

$$\hat{K}_X^\tau = G_K(u_{X,K}^\tau), \qquad \hat{E}_X^\tau = G_E(u_{X,E}^\tau),$$

where $\hat{K}_X^\tau \in \mathbb{R}^{N_K\times 2}$ denotes predicted object-centric point locations, and $\hat{E}_X^\tau \in \mathbb{R}^2$ denotes the predicted wrist position for human videos or end-effector position for robot videos.

Let $K_X^\tau \in \mathbb{R}^{N_K \times 2}$ and $E_X^\tau \in \mathbb{R}^2$ be the corresponding annotations. We denote their visibility masks as $M_{X,K}^{\tau,k} \in \{0,1\}$ and $M_{X,E}^\tau \in \{0,1\}$. Using the Huber loss $\mathcal{H}(\cdot,\cdot)$, the masked keypoint loss is

$$\ell_K(X) = \frac{\sum_\tau \sum_{k=1}^{N_K} M_{X,K}^{\tau,k}\, \mathcal{H}\left(\hat{K}_{X,k}^\tau, K_{X,k}^\tau\right)}{\sum_\tau \sum_{k=1}^{N_K} M_{X,K}^{\tau,k} + \epsilon},$$

and the masked wrist/end-effector loss is

$$\ell_E(X) = \frac{\sum_\tau M_{X,E}^\tau\, \mathcal{H}\left(\hat{E}_X^\tau, E_X^\tau\right)}{\sum_\tau M_{X,E}^\tau + \epsilon}.$$

The auxiliary loss is

$$\ell_{\text{aux}}(X) = \lambda_K \ell_K(X) + \lambda_E \ell_E(X),$$

and the batch-level objective is

$$\mathcal{L}_{\text{aux}} = \frac{1}{|\mathcal{B}|} \sum_{X \in \mathcal{B}} \ell_{\text{aux}}(X).$$

Here, ϵ is a small constant for numerical stability.

Source-relative pair-discriminative alignment loss. The alignment loss is applied to paired human-robot demonstrations. For a paired batch $\mathcal{B}_p = \{(H_i, R_i)\}_{i=1}^B$, we define

$$f_i^H = \rho(Z_{H_i}) = \rho(Z_{H_i 0}), \qquad f_i^{R0} = \rho(Z_{R_i 0}), \qquad f_i^R = \rho(Z_{R_i}),$$

where f_i^H is the frozen human representation, f_i^{R0} is the frozen robot representation, and f_i^R is the adapted robot representation. All features are ℓ_2-normalized before computing cosine distances. We use

$$d(u,v) = 1 - \cos(u,v), \qquad d_i^+ = d(f_i^R, f_i^H).$$

Source-relative term. The source-relative term requires the adapted robot feature to improve over the frozen robot feature with respect to the paired human feature:

$$\ell_{\text{SR}}(i) = \left[m_s + d_i^+ - d(f_i^{R0}, f_i^H)\right]_+,$$

where $[x]_+ = \max(x, 0)$, and m_s is the source-relative margin.

Pair-discriminative term. To maintain pair-level discrimination, we use non-matching pairs in the batch as negatives. For $B > 1$, the mean negative distances in the two matching directions are

$$\bar{d}^{R\to H}(i) = \frac{1}{B-1} \sum_{j \neq i} d(f_i^R, f_j^H), \qquad \bar{d}^{H\to R}(i) = \frac{1}{B-1} \sum_{j \neq i} d(f_j^R, f_i^H).$$

The corresponding pair-discriminative losses are

$$\ell_\alpha(i) = \left[m_t + d_i^+ - \bar{d}^\alpha(i)\right]_+, \qquad \alpha \in \{R \to H,\ H \to R\},$$

where m_t is the pair-discrimination margin. We summarize the two directions as

$$\ell_{\text{PD}}(i) = \lambda_{R\to H} \ell_{R\to H}(i) + \lambda_{H\to R} \ell_{H\to R}(i).$$

The alignment loss for pair i is

$$\ell_{\text{align}}(i) = \ell_{\text{SR}}(i) + \ell_{\text{PD}}(i).$$

The batch-level alignment loss is

$$\mathcal{L}_{\text{align}} = \frac{1}{|\mathcal{B}_p|} \sum_{i=1}^{|\mathcal{B}_p|} \ell_{\text{align}}(i).$$

When $B = 1$, the pair-discriminative term is omitted and only the source-relative term is used.

The full Stage-1 objective is

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{lam}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}.$$

A.3 Model Architecture

Vision encoder with robot-only adapter. We follow Prismatic-7B [37] and use a fused vision encoder including DINOv2 [38] SigLIP [39] to obtain vision features. By complementing the features of different visual backbones, the generalization and fine-grainedness of visual representations can be improved.

For robot-only adatper, we append a 2-layer MLP after every attention and FFN block in each layer of DINOv2 and SigLIP. Given the current feature as input, the MLP predicts a residual term that is added back to the original feature. The first MLP layer is Gaussian-initialized, while the second layer is zero-initialized, ensuring the output features are identical to the original features at initialization.

Latent action model architecture.

Table A2: LAM configuration used in Stage-1 training.

Component	Configuration
Visual backbone	DINOv2 + SigLIP fused patch tokens
Input image size	224×224
Visual tokens per frame	$16 \times 16 = 256$
Fused token dimension	2176
Transition horizon	random sampled
Latent-action tokens per transition	$N_q = 4$
VQ codebook size	$K = 16$
Latent dimension	$d_q = 128$
VQ commitment weight	$\beta = 0.25$
Text encoder	frozen T5 encoder
Auxiliary tokens	1 keypoint token, 1 end-effector token

We follow UniVLA [2] to construct the latent action model. Given an input video clip and a task instruction, the model first converts each frame into visual patch tokens and encodes the language with a text encoder. These visual and language tokens are then fed into a spatiotemporal transformer together with several learned latent query tokens. Among them, the action queries are responsible for capturing the underlying dynamics of the sequence, while additional queries are used to model structured information such as object keypoints and end-effector states.

The core idea of the architecture is to compress action-relevant temporal information into a small set of quantized latent action codes through a vector-quantization bottleneck. These discrete latent codes serve as a high-level abstraction of the motion or behavior needed to explain the video sequence under the given instruction. A decoder then combines the latent action codes with visual context to reconstruct future visual features, while auxiliary heads predict keypoints and end-effector positions to provide extra structural supervision.

VLA architecture. We follow OpenVLA-OFT [28] to construct the generalist robot policy backbone, which is built upon Prismatic-7B [37]. The architecture contains the same fused vision encoder including SigLIP [39] and DINOv2 [38] in latent action model, a projector aligning fused visual and language embedding space, and a LLaMA-2 [40] LLM backbone. An extra action head is employed in finetuning stages, following OpenVLA-OFT.

Building upon OpenVLA-OFT, we incorporate the robot-only adapter from HARP-LAM into the vision encoder of PrismaticVLM using the same design. Since the fused vision encoder in PrismaticVLM is frozen during web-scale pretraining, the copied weights preserve the same representation alignment properties after initialization.

Moreover, web-scale VLM pretraining is dominated by human-centric data rather than robot data, yielding more semantically robust human representations. Under the HARP framework, the human representation space remains fixed while robot representations are aligned toward the human domain. Consequently, the original alignment between the VLM vision encoder and the LLM is largely preserved, making such initialization well-suited for subsequent VLA training.

A.4 Human-Robot Representation and Latent-Action Alignment: Experiment Details

We evaluate representation and latent-action alignment on 93 held-out paired human-robot demonstrations that are not used for training. For all methods, we use the same frame sampling strategy, video-level pooling function $\rho(\cdot)$, feature normalization, and cosine distance/similarity metrics. Unless otherwise specified, we use the fused DINOv2+SigLIP feature branch. Patch tokens are first averaged spatially within each frame and then temporally averaged over sampled frames to obtain a video-level embedding, followed by ℓ_2-normalization.

UMAP visualization. For visual-representation UMAP, we compare the original feature space $F(H)$ versus $F(R)$ with the adapted feature space $F(H)$ versus $T_\theta(R)$. Human and robot em-

beddings from the same held-out paired set are projected with UMAP using cosine distance. Circles denote human videos and triangles denote robot videos; gray points show sampled pairs and colored points highlight a small subset of pairs for readability. For latent-action UMAP, we extract the quantized latent-action embeddings from HARP-LAM. The baseline panel uses latent actions computed from the original visual encoder for both human and robot videos, while the HARP-LAM panel uses the frozen human branch and the adapted robot branch. The latent-action tokens are flattened and ℓ_2-normalized before UMAP projection.

Paired cosine distance. For paired cosine distance, we compute

$$d_i^{\text{orig}} = 1 - \cos(\rho(F(H_i)), \rho(F(R_i)))$$

for the unadapted encoder, and

$$d_i^{\text{adapt}} = 1 - \cos(\rho(F(H_i)), \rho(T_\theta(R_i)))$$

for adapted encoders. Fig. 5 visualizes the distribution of pair-level distances over the 93 held-out pairs using box plots, where lower values indicate smaller human-robot representation gaps.

Cross-embodiment retrieval. For bidirectional retrieval, H2R uses each human video as a query and ranks all 93 robot videos by cosine similarity, while R2H uses each robot video as a query and ranks all 93 human videos. The matched demonstration is the diagonal positive pair in the similarity matrix. We report Recall@1 in the main paper and provide additional retrieval metrics, including mean reciprocal rank (MRR), in Table A3.

Table A3: Supplementary bidirectional cross-embodiment retrieval metrics on held-out paired demonstrations.

Method	H2R R@1	H2R MRR	R2H R@1	R2H MRR	Avg. R@1	Avg. MRR
Unadapted	44.09	0.5936	43.01	0.5882	43.55	0.5909
HR	45.16	0.6022	45.16	0.6269	45.16	0.6145
HARP-HR	46.24	0.6286	60.22	0.7398	53.23	0.6842
HARP-L2	70.97	0.8310	52.69	0.6742	61.83	0.7526
HARP-SR	84.95	0.9091	64.52	0.7680	74.74	0.8385
HARP-SRPD	**87.10**	**0.9283**	**69.89**	**0.8010**	**78.50**	**0.8647**

Cross-embodiment latent-action correspondence. For latent-action retrieval, we extract quantized latent-action embeddings for temporally aligned human and robot transitions from the same held-out paired pool. The latent-action tokens are flattened and ℓ_2-normalized. H2R retrieval uses each human transition as a query to rank robot transitions by cosine similarity, while R2H is defined symmetrically, with the temporally corresponding transition treated as the positive. We report the average H2R/R2H Recall@1 as Lat. R@1.

We additionally measure discrete code agreement (CA). For each temporally corresponding human-robot transition, CA is the fraction of the N_q latent-action slots assigned the same VQ code. To discount accidental agreement caused by global code usage, we randomly shuffle robot samples across human samples and recompute the agreement. We define

$$\text{CA-gap} = \text{CA}_{\text{paired}} - \text{CA}_{\text{shuffled}}.$$

A larger CA-gap indicates stronger correspondence-specific agreement of discrete latent actions across embodiments.

A.5 Component Ablation and Paired-Data Scaling Details

Component ablations. We perform controlled Stage-1 ablations while keeping all other training and evaluation settings fixed. For paired samples, Self-only uses only self-prediction, Cross-only uses only cross-prediction, and Full HARP combines both; unpaired samples always use self-prediction. Paired-only removes unpaired training videos, while w/o Aux removes object-keypoint and wrist/end-effector supervision. Across these variants, codebook perplexity remains 10.0–14.2 out of a maximum of 16, indicating substantial codebook utilization rather than collapse.

Paired-data scaling. We subsample the unique paired human-robot demonstrations to 25% and 5% of the full paired set, corresponding to approximately 6M and 1.2M processed paired frames, respectively. All unpaired videos, data-mixture settings, optimization budget, preprocessing, and evaluation sets are kept fixed. Vis. R@1 and Lat. R@1 use the same held-out paired evaluation pool

as Sec. 4.1, while RLBench and CALVIN follow the same downstream protocols as the full-data model.

Controlled comparison. For controlled HARP variants, all methods use the same paired/unpaired training split, DTW preprocessing, auxiliary cues, visual backbone, robot-only adapter, and training budget; only the alignment objective or pooling space differs. HR-Align baselines use the same fused backbone and robot-only adapter but follow the original paired-data alignment setup.

A.6 Robot Policy Learning with Aligned Visual Representations: Experiment Details

We follow the RLBench evaluation protocol used by HR-Align[8] and evaluate frozen visual encoders on 18 RLBench manipulation tasks. To isolate the effect of visual representation quality, all methods use the same downstream policy architecture, policy head, action space, demonstration data, training schedule, and optimization budget. The only component changed across methods is the frozen visual encoder used to extract visual representations.

For each method, the corresponding visual encoder is initialized from the evaluated representation-learning checkpoint and kept frozen during downstream policy learning. The policy head receives frozen visual features and predicts robot actions under the same imitation-learning objective for all methods. Each task is evaluated over 75 episodes using the standard RLBench success signal, resulting in 1,350 evaluation episodes per method. We report the average success rate across the 18 tasks in Table 4.

The evaluated RLBench tasks are: `put_item_in_drawer`, `reach_and_drag`, `turn_tap`, `slide_block_to_color_target`, `open_drawer`, `put_groceries_in_cupboard`, `place_shape_in_shape_sorter`, `put_money_in_safe`, `push_buttons`, `close_jar`, `stack_blocks`, `place_cups`, `place_wine_at_rack_location`, `light_bulb_in`, `sweep_to_dustpan_of_size`, `insert_onto_square_peg`, `meat_off_grill`, and `stack_cups`.

A.7 VLA Baselines and Real-world Task Details

Evaluation on CALVIN Benchmark. The CALVIN benchmark [27] is a language-conditioned long-horizon robotic manipulation benchmark designed to evaluate multi-task and sequential task execution. It consists of a tabletop manipulation environment with diverse objects and scenes, where agents must complete sequences of instructions conditioned on visual observations and natural language commands. Following prior work, we evaluate policies on the most challenging CALVIN ABC→D setting, where robots are trained with standard datasets collected from environments ABC and tested in the unseen environment D. For CALVIN, Task1–Task5 report the percentage of trials that complete at least the first k subtasks in a five-instruction sequence, and Avg. Len. denotes the average number of completed subtasks.

Evaluation on Real-world Settings. Our policy uses an 18-dimensional action space, consisting of 6D end-effector control for the XArm7 and 12D joint control for the XHand. The policy takes RGB observations from a third-view camera and a wrist-mounted camera as visual inputs, as well as language instruction, as shown in Fig A2.

To evaluate real-world performance, we choose a set of comprehensive manipulation tasks, including: pick and place, push object, press button, and flip cup.

- Pick and Place: the robot was asked to place a specified item into a plate.
- Push Object: the robot was asked to push a designated item forward.
- Press Button: the robot was asked to press a button.
- Flip Cup: the robot was asked to flip the cup upside down and stand it upright.

These tasks test HARP-VLA's language grounding, spatial understanding, and dexterity capabilities. Each method is tested for 60 trials per task under the same task initialization and success criteria, and we report success rates. All HARP-VLA variants use the same downstream real-action demonstrations, action space, action chunk length, and finetuning schedule; they differ only in the pretrained initialization and whether the vision encoder is frozen during VLA pretraining.

Baselines. We selected 5 representative baselines as below, among which UniVLA is similar to our training method, while OpenVLA-OFT is similar to our architecture. The difference between OpenVLA-OFT and ours is that we freeze the human-robot aligned vision encoder and pre-train only on video data, while OpenVLA-OFT is pre-trained on labeled robot data. For baselines' results, we

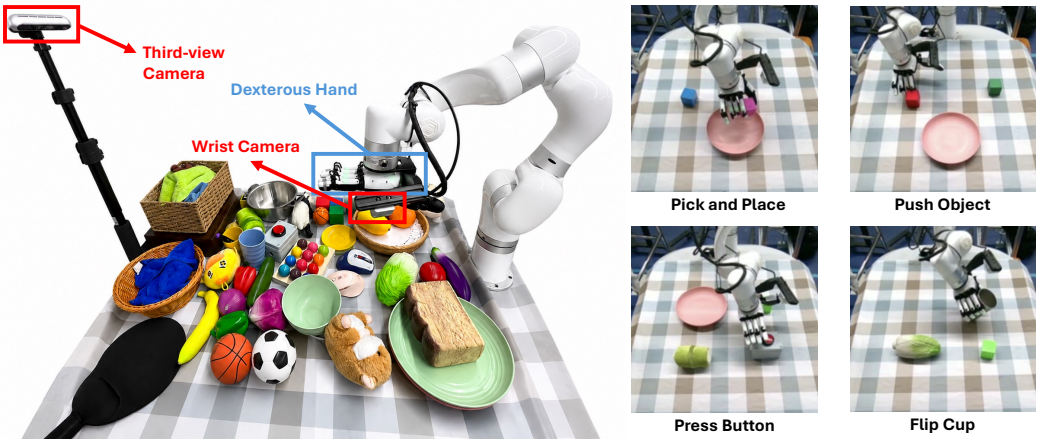

Figure A2: Real-world platform setup and task overview

use reproduced or official results under the corresponding benchmark protocols. When reproducing baselines, we use the same downstream demonstrations and evaluation settings as HARP-VLA.

- **OpenVLA** [11] is an open-source VLA that learns manipulation skills from robotic datasets by predicting discrete action tokens jointly conditioning on visual observations and natural language instructions.
- **UniVLA** [2] learns transferable task-centric latent actions from large-scale videos to enable cross-embodiment pre-training for VLAs.
- **OpenVLA-OFT** [28] proposes an efficient fine-tuning strategy that improves the inference speed and performance of OpenVLA [11].
- $\boldsymbol{\pi_0}$ [14] is a generalist VLA that generates continuous robot actions via flow-matching.
- $\boldsymbol{\pi_{0.5}}$ [15] improves π_0 with enhanced training and scaling to better support open-world robotic manipulation.

Following the settings of the original paper itself, in our stage-3 experimental settings, OpenVLA and UniVLA uses third-view image as visual input, and OpenVLA-OFT, π_0, $\pi_{0.5}$ uses third-view and wrist view images, which is consistent with HARP. Except for OpenVLA, which directly outputs the current action, all other baselines output an action chunk of length 10, which is the same as HARP settings.
For the CALVIN experiment, we finetuned π_0, $\pi_{0.5}$, and Openvla-OFT ourselves, and reported the OpenVLA and UniVLA results using publicly available data. For the real-world experiment, we finetuned all of them.