Cross-Hand Latent Representation for Vision-Language-Action Models

Guangqi Jiang[1*] Yutong Liang[1*] Jianglong Ye[1] Jia-Yang Huang[1] Changwei Jing[1]
Rocky Duan[2] Pieter Abbeel[2,3] Xiaolong Wang[1†] Xueyan Zou[1†]
[1]UC San Diego [2]Amazon FAR [3]UC Berkeley * Equal Contribution †Equal Advising
https://xl-vla.github.io

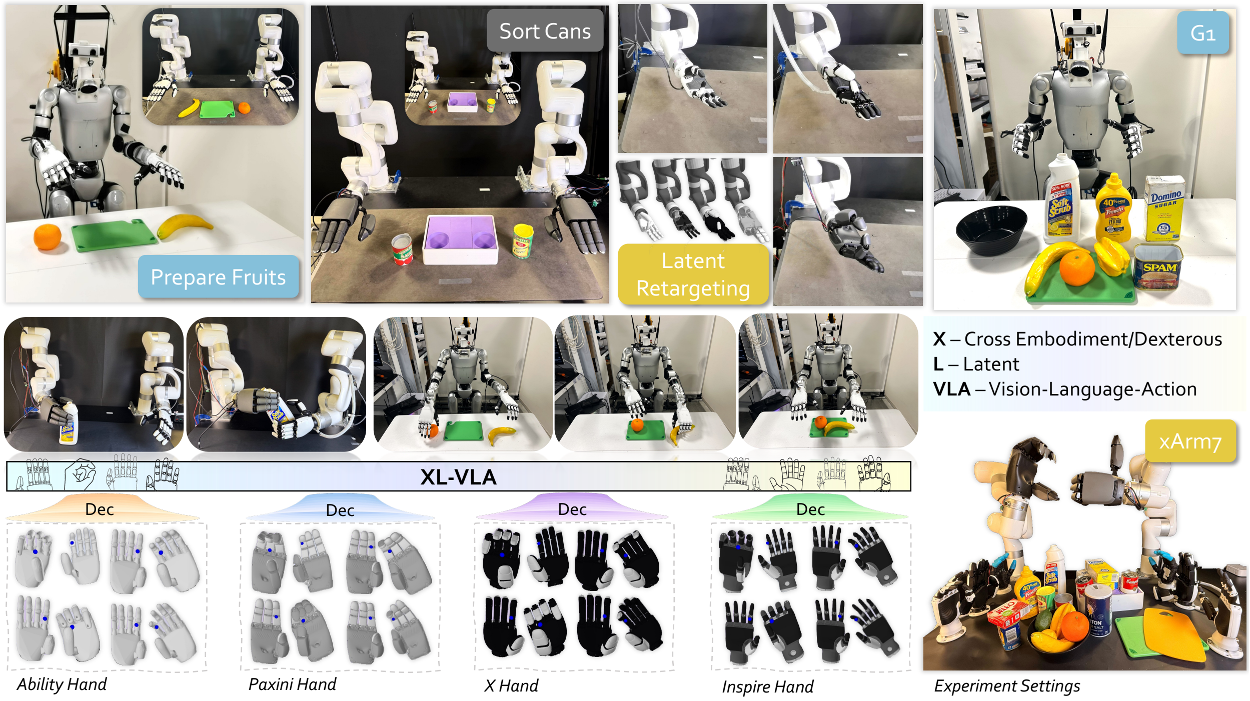


Figure 1. **Overview.** XL-VLA enables direct decoding of a single latent action into multiple dexterous hand embodiments. Shown above, an action prediction can be instantiated on the Ability hand, Paxini DexH13 hand, X-Hand1, and Inspire hand for language-guided manipulation. We show our experiment settings on the right figure with collected objects and DexHands.

Abstract

Dexterous manipulation is essential for real-world robot autonomy, mirroring the central role of human hand coordination in daily activity. Humans rely on rich multimodal perception—vision, sound, and language-guided intent—to perform dexterous actions, motivating vision-based, language-conditioned manipulation systems for robots. However, training reliable vision-language-action (VLA) models for dexterous manipulation requires large-scale demonstrations across many robotic hands. In addition, as new dexterous embodiments appear rapidly, collecting data for each becomes costly and impractical, creating a need for scalable cross-embodiment learning. We introduce XL-VLA, a vision-language-action framework integrated with a unified latent action space shared across diverse dexterous hands. This embodiment-invariant latent space is directly pluggable into standard VLA architectures, enabling seamless cross-embodiment training and efficient reuse of both existing and newly collected data. Experimental results demonstrate that XL-VLA consistently outperforms baseline VLA models operating in raw joint spaces, establishing it as an effective solution for scalable cross-embodiment dexterous manipulation.

1. Introduction

Recent progress in vision-language-action (VLA) modeling has begun to extend the successes of large-scale vision and language models into robotics, enabling robots to interpret visual scenes, follow natural language instructions, and execute complex behaviors in the physical world. A key insight behind these advances is that unifying vision and language can be naturally expressed

Paper	**Data**	**Deployment**		**Input**			**Output**	
	Type(s)	**Embodiment**	**EEF ↔ EEF**	**Vision**	**Lang**	**Prop**	**Decoder**	**ZS**
UniVLA [8]	human video; teleoperation	1 arm+1 gripper	gripper ↔ gripper	✓	✓	✗	✓	✗
ATE [71]	teleoperation; simulation	2 arms+2 grippers	—	✓	✓	✗	✗	✗
LAD [3]	teleoperation	1 arm+1 hand/gripper	hand ↔ hand/gripper	✓	✗	✗	✓	✗
EgoBridge [45]	human video; teleoperation	2 arms+2 pushers	—	✓	✗	✓	✗	✗
CoMo [65]	internet video; teleoperation	1 arm+1 gripper	—	✓	✗	✗	✗	✗
Tenma [17]	teleoperation	2 arms+2 grippers	—	✓	✓	✓	✗	✗
CycleVAE [16]	teleoperation	1 arm+1 hand	hand ↔ hand	✗	✗	✓	✓	✗
CETransfer [53]	simulation (sim → real)	1 arm+1 gripper	gripper ↔ gripper	✗	✗	✓	✓	✓
Ours	teleoperation	2 arm+2 hand	hand ↔ hand	✓	✓	✓	✓	✓

Table 1. **Related Work Summary.** Summary of related work comparing data sources, deployment settings, and input/output capabilities for latent-based cross-embodiment methods. Data indicates the training modalities used in each work. Deployment specifies the robot embodiments evaluated and whether cross–end-effector transfer is supported. Input denotes which modalities (vision, language, proprioception) are used for training. Output reports whether a method includes a cross-embodiment decoder and whether it enables zero-shot transfer to unseen embodiments.

through sequence-to-sequence modeling, and VLA systems can adopt the same abstraction by treating actions as an additional output modality.

However, a fundamental obstacle emerges when moving from vision and language to action: while language possesses a relatively stable and universal vocabulary, robotic action spaces are inherently tied to the morphology of the robot. For dexterous hands in particular, action parameterizations—joint positions—vary significantly across embodiments and continue to evolve rapidly with new hardware designs. This raises two key questions for scalable robot learning: (1) *How can we define a unified action representation within a family of robots?* (2) *How can we seamlessly integrate a new robot whose action space differs from existing ones?*

In this work, we address these challenges by introducing a *shared latent action space* tailored for **dexterous hands**. This latent space serves as an embodiment-invariant representation that enables joint training across heterogeneous hands. While prior VLA and cross-embodiment efforts have primarily focused on robotic arms equipped with parallel grippers, we focus on the substantially more complex, and more capable domain of dexterous manipulation. Moreover, we emphasize *real-world* datasets and physical robot evaluation, demonstrating that our method remains robust under significant cross-embodiment variation.

We summarize our contributions as follows:

- We collect a large-scale teleoperation dataset covering 10 manipulation tasks across four newly introduced dexterous hands—Ability, Paxini DexH13, X-Hand1, and Inspire—containing 2M state-action pairs.
- We propose an unsupervised latent autoencoder framework that learns a unified action space applicable to a wide range of hands.
- We introduce XL-VLA, a full VLA pipeline built upon the cross-embodiment latent action space. XL-VLA achieves significantly stronger cross-embodiment performance than standard VLA baselines and exhibits **zero-shot generalization** to untrained cross-embodiment task configurations.

2. Related Work

Dexterous Manipulation. The direction of dexterous manipulation focuses on utilizing DexHand for standard manipulation tasks, aiming to enable more complex operations. This field encompasses various areas of focus, including manipulator hardware [34, 48, 67], sensors [51, 63], learning and control algorithms [12, 28, 31, 40], and human-robot interaction [13, 23, 61]. In this work, we specifically concentrate on learning and control algorithms, leveraging vision-language-action (VLA) models [30, 36, 46, 50, 69]. Furthermore, we define a unified action space to support cross-embodiment dexterous manipulation [24, 44, 55].

Cross Embodiment. Cross embodiment typically refers to learning a *single* policy that can flexibly adapt across diverse embodiments—e.g., different humanoids or dexterous hands—without per-robot retraining [21, 22, 25, 26, 43, 52, 66, 74]. Within this area, approaches leverage human video for supervision [15, 32, 42, 45, 47, 70], apply imitation learning with motion retargeting to bridge morphology gaps [11, 39, 49, 59], and employ generative models to synthesize action-consistent trajectories across bodies [1, 3, 27, 57, 75]. A complementary line of work constructs unified latent action spaces that factor out embodiment-specific details, enabling transfer and zero-shot reuse across platforms [3, 9, 17, 54, 71, 73]. This paper follows the latter paradigm, aligning actions in a shared representation for robust cross-embodiment control.

Hand/Dex Retargeting. Hand retargeting for teleoper-

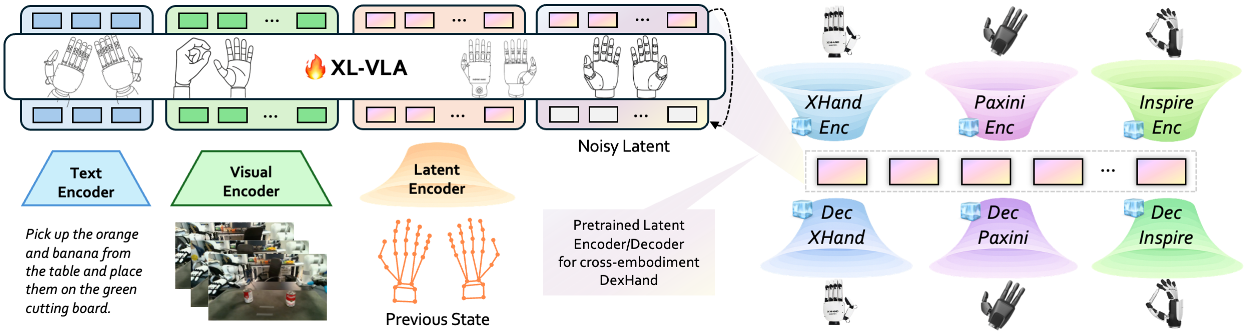

XL-VLA
Text Encoder
Visual Encoder
Latent Encoder
Noisy Latent
Pick up the orange and banana from the table and place them on the green cutting board.
Previous State
Pretrained Latent Encoder/Decoder for cross-embodiment DexHand
XHand Enc
Paxini Enc
Inspire Enc
Dec XHand
Dec Paxini
Dec Inspire

Figure 2. **Model Pipeline.** XL-VLA builds on π_0 [6] with vision and language encoders paired with an action expert that operates in a shared latent action space for cross-embodiment control. During VLA training, the action expert is finetuned while the pretrained latent encoders and decoders remain frozen.

ation and imitation learning has progressed from kinematic pipelines to fast, principled learning: GeoRT delivers 1 kHz unsupervised mapping [68]; contact-aware and unified formulations improve human–object fidelity [35, 37], with objective ablations [60] and practical systems spanning real-time teleop and hardware-agnostic platforms [14, 19, 58]. Beyond copying humans, functional and policy-centric retargeting improves task outcomes [41, 62], while type-guided teleop exploits robot-specific dexterity [38].

Latent Action Space. As shown in Table. 1, latent action spaces provide embodiment agnostic control codes that align input modalities (vision, language, proprioception) and decode to diverse robots. Examples span discrete VQ tokens with per-robot decoders [8], continuous end-effector latents trained on retargeted pairs and generated by diffusion [3], and unified action VAEs that steer existing VLA policies [71]. Other directions align policy features or motion rather than a single EEF space optimal-transport co-training [45], Internet-video motion embeddings [65], diffusion transformers with standardized tokens [17] and cycle/adversarial mappings enabling cross-embodiment decoding and sim→real transfer [16, 53].

Vision-Language-Action Model. VLA models adapt large vision–language models (VLMs) to robot control by discretizing actions and predicting them autoregressively, enabling transfer of web-scale priors to manipulation [7, 29, 33, 56]. While these systems demonstrate broad generalization, their tokenized action decoding can hinder high-rate, dexterous control [72]. In contrast, we fine-tune a pre-trained VLM backbone initialized from PaliGemma [5] on teleoperated trajectories. Our action expert regresses continuous *latent action chunks*: each target is represented by a single latent vector produced by our hand-specific encoder. During training, we replace π_0's original state tokens with these latent tokens and finetune on the next latent chunk, allowing a single hand-agnostic VLA policy to operate across multiple dexterous hands while preserving the benefits of VLM pretraining.

3. Method

3.1. Preliminary

Problem Formulation. In this work, we address the problem of *language-guided cross-embodiment dexterous manipulation* based on visual perception. For a dexterous hand $h \in \mathcal{H}$ with d_h actuated joints we control absolute joint rotations $\mathbf{q}^{(h)} \in \mathbb{R}^{d_h}$. At the policy level we operate on *action chunks*: each action $\mathbf{q}_t^{(h)} \in \mathbb{R}^{64\times d_h}$ is a sequence of 64 joint-position commands sampled at 20 Hz (3.2 s of motion). At control step t, the policy receives a short history of joint states, the previously executed action chunk $\mathbf{q}_t^{(h)}$, the current image observations $\mathbf{V}$, and a language instruction $\mathbf{T}$, and predicts the next chunk $\mathbf{q}_{t+1}^{(h)}$ via $\mathbf{q}_{t+1}^{(h)} = F(\mathbf{q}_t^{(h)}, \mathbf{V}, \mathbf{T})$.

The objective is to predict embodiment-consistent joint-rotation trajectories conditioned on these multimodal inputs with a unified multi-task VLA model. Although the continuous joint spaces $\mathbf{q}^{(h)}$ are hand-specific, the sequence model F itself is hand-agnostic; the hand identity h is used only to choose the appropriate encoder/decoder that maps between joint space and the shared latent action space described below. To evaluate this setting, we consider a diverse set of dexterous robotic hands, including the *Ability Hand*, *Inspire Hand*, *X-Hand1*, and *Paxini DexH13*, which vary in structure, actuation, and kinematics.

To tackle this problem, we introduce an embodiment-invariant latent action space that integrates seamlessly into a vision–language–action (VLA) framework. This latent space provides a unified representation across diverse dexterous hands, enabling the model to train effectively on cross-embodiment data and generalize manipulation skills beyond a single hand morphology. Furthermore, the proposed latent space supports transferring control policies across different embodiments without requiring hand-specific retraining.

Pipeline. As illustrated in Fig. 2, our proposed framework consists of two main components: (1) a VLA backbone that encodes multimodal inputs $(\mathbf{V}, \mathbf{T})$, and (2) a pretrained set of latent encoders and decoders designed for cross-embodiment transfer. Our VLA design follows π_0 [6], which employs vision and language encoders together with an action expert. In the original π_0, proprioceptive history is provided through a stack of state tokens. In XL-VLA we instead feed *latent action tokens*: for each hand h, a hand-specific encoder E_h maps the previous absolute joint-position action chunk $\mathbf{q}_t^{(h)}$ (64 frames at 20 Hz) into a compact latent vector $\mathbf{z}_t = E_h(\mathbf{q}_t^{(h)})$. The VLA model conditions on a short history of such latent tokens, together with vision and language tokens, and predicts the next latent chunk $\widehat{\mathbf{z}}_{t+1}$. This latent is decoded by the embodiment-specific decoder D_h to obtain the next joint command chunk $\widehat{\mathbf{q}}_{t+1}^{(h)} = D_h(\widehat{\mathbf{z}}_{t+1})$. During VLA finetuning we keep all latent encoders and decoders frozen.

This embodiment-invariant latent representation $\mathbf{z}$ acts as a unified action space shared across heterogeneous dexterous hands. By learning and decoding actions within this latent space, the model effectively bridges differences in morphology and actuation across embodiments, enabling a single hand-agnostic VLA policy to operate on diverse robotic hands. The hand identity h is used only to select the appropriate encoder E_h and decoder D_h and is never provided as an explicit input token to the VLA backbone. We describe the detailed design and training of this latent action space in the following sections.

3.2. Latent Space

Definition. Rather than defining a separate action space for each dexterous hand, we introduce a *shared latent action space* that provides a unified representation for all dexhand embodiments. This latent space is pretrained independently of the VLA model through a set of hand-specific encoders and decoders that all map to the same latent distribution. As a result, the latent embedding acts as an implicit, embodiment-agnostic action space that can be used by downstream policies to seamlessly control different dexterous hands.

3.2.1. Modeling

To construct the latent representation, we employ a multi-headed VAE-style autoencoder. For each hand type $h \in \mathcal{H}$ (e.g., X-Hand, Ability, Inspire, Paxini), we define a hand-specific encoder E_h and decoder D_h. Each hand provides a joint configuration $\mathbf{q}^{(h)} \in \mathbb{R}^{d_h}$, where $\mathbf{q}^{(h)}$ denotes the joint position values (q-pos) and d_h is the dimensionality of that embodiment. The encoder outputs the parameters of a Gaussian posterior $(\boldsymbol{\mu}^{(h)}, \boldsymbol{\sigma}^{(h)}) = E_h(\mathbf{q}^{(h)})$, from which we sample a latent code $\mathbf{z}$ using the reparameterization trick, $q(\mathbf{z} \mid \mathbf{q}^{(h)}) = \mathcal{N}(\boldsymbol{\mu}^{(h)}, \mathrm{diag}((\boldsymbol{\sigma}^{(h)})^2))$. The decoder reconstructs back into the corresponding joint space $\hat{\mathbf{q}}^{(h)} = D_h(\mathbf{z})$.

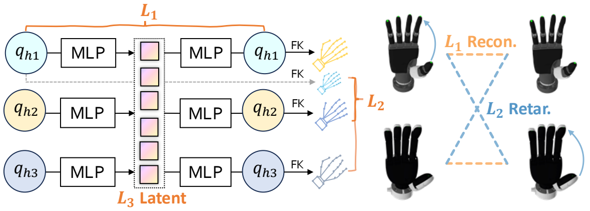


Figure 3. **Latent space pretraining pipeline.** For each hand type, joint positions $\mathbf{q}_h$ are mapped through an encoder MLP into a shared latent space and reconstructed by a decoder MLP. The diagram also indicates the placement of the reconstruction loss L_1, retargeting loss L_2 via differentiable forward kinematics, and latent regularization loss L_3.

In practice, each encoder and decoder is implemented as a lightweight MLP: the input q-pos vector $\mathbf{q}^{(h)}$ is projected into a common latent space through the encoder MLP, and the decoder MLP reprojects the latent embedding back into the hand's original joint configuration. This architecture provides a unified latent manifold while preserving the structure of each embodiment. To shape a meaningful cross-embodiment latent space, we impose three training constraints: (1) a *reconstruction constraint* L_1 ensuring $\hat{\mathbf{q}}^{(h)}$ matches $\mathbf{q}^{(h)}$, (2) a *retargeting constraint* L_2 aligning fingertip geometry across hands using differentiable forward kinematics, and (3) a *latent constraint* L_3 regularizing the latent embedding to follow a smooth prior distribution. Together, these constraints encourage the latent space to capture embodiment-invariant structure, enabling consistent decoding into any dexterous hand.

3.2.2. Objective

Reconstruction Loss (L_1). Since each DexHand embodiment has its own joint space, we first require that the hand-specific encoder-decoder pair behaves as an autoencoder on that hand. Given a joint configuration $\mathbf{q}^{(h)}$ and its reconstruction $\hat{\mathbf{q}}^{(h)}$ for hand $h \in \mathcal{H}$, the reconstruction loss averaged over all hands is

$$L_1 = \mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \mathrm{MSE}\Big(\hat{\mathbf{q}}^{(h)}, \mathbf{q}^{(h)}\Big), \qquad (1)$$

which ensures that the latent space preserves hand-specific kinematics and that no embodiment is degraded by sharing the latent representation.

Retargeting Loss (L_2). To make the latent space truly cross-embodiment, we align fingertip geometry between different DexHand robots. For each hand h, we use differentiable forward kinematics (FK) to map joints to fingertip positions $\mathbf{p}_i^{(h)}$, and define fingertip displacements $\boldsymbol{\delta}_{ij}^{(h)} = \mathbf{p}_i^{(h)} - \mathbf{p}_j^{(h)}$ for fingertip pairs $(i, j) \in \mathcal{P}$.

Method	Hand	PF	SC	SoC	HB	RL	PS	RB	PuS	PoS	PC	Mean
π_0 [6]	Ability	0.10	0.10	0.00	0.70	0.20	0.80	0.60	0.30	0.30	0.60	0.37
	Inspire	0.10	0.20	0.00	0.30	0.10	0.50	0.30	0.20	0.20	0.80	0.27
	Paxini	0.40	0.40	0.30	0.20	0.00	0.80	0.60	0.30	0.10	0.40	0.35
	XHand	0.20	0.40	0.00	0.40	0.10	0.60	0.30	0.20	0.30	0.40	0.29
	Mean	0.20	0.28	0.08	0.40	0.10	0.68	0.45	0.25	0.23	0.55	0.32
XL-VLA	Ability	**0.80**	**0.80**	0.40	**1.00**	0.70	**1.00**	0.70	0.30	0.90	0.70	0.73
	Inspire	0.60	0.50	0.50	0.80	0.40	0.80	**1.00**	**0.40**	0.80	**1.00**	0.68
	Paxini	**0.80**	0.70	**0.80**	**1.00**	0.30	**1.00**	**1.00**	**0.40**	0.80	**1.00**	**0.78**
	XHand	0.60	0.50	0.50	**1.00**	0.30	**1.00**	0.90	0.30	**1.00**	0.90	0.70
	Mean	0.70+50%	0.63+35%	0.55+47%	0.95+55%	0.43+33%	0.95+27%	0.90+45%	0.35+10%	0.88+65%	0.90+35%	0.72+40%

Table 2. **Vision-Language-Action Modeling.** We compare XL-VLA with π_0 under cross-embodiment training. Although π_0 can handle different embodiments by adjusting sequence length, our method achieves consistently higher success rates across all hands and tasks. The first row PF, SC, SoC, etc. denote each task introduced in Task & Dataset, and each task is executed for 10 times, and we compute the success rate for each task cross hand.

The pair set $\mathcal{P}$ contains thumb–finger pairs for the four aligned digits (thumb–index, thumb–middle, thumb–ring, thumb–little). We manually align finger indices across hands so that these digits correspond semantically; for Paxini DexH13, which lacks a little finger, we drop any pairs involving that digit when evaluating L_2. The retargeting loss penalizes discrepancies in pinch distances and directions between source hands s and target hands t:

$$L_2 = \frac{1}{|\mathcal{H}|(|\mathcal{H}|-1)|\mathcal{P}|} \sum_{s\neq t} \sum_{(i,j)\in\mathcal{P}} w_{ij}^{(s)} \Big[\lambda_{\text{dis}} \big(\|\boldsymbol{\delta}_{ij}^{(s)}\|_2 - \|\hat{\boldsymbol{\delta}}_{ij}^{(t)}\|_2\big)^2 + \lambda_{\text{dir}} \big(1 - c_{ij}^{(s,t)}\big)\Big]. \tag{2}$$

where $\hat{\boldsymbol{\delta}}_{ij}^{(t)}$ is computed from the decoded configuration of hand t, $c_{ij}^{(s,t)}$ denotes the cosine of the angle between the pinch directions $\boldsymbol{\delta}_{ij}^{(s)}$ and $\hat{\boldsymbol{\delta}}_{ij}^{(t)}$, and $w_{ij}^{(s)} = \exp(-\lambda_{\text{dis}}^{\text{exp}}\|\boldsymbol{\delta}_{ij}^{(s)}\|_2)$ emphasizes tighter pinches. This loss encourages the same latent code to produce geometrically consistent pinch behaviors across different hands.

Latent Loss (L_3). Finally, we regularize the DexHand latent space to be smooth and well-behaved by imposing a standard Gaussian prior on the latent variables. For the approximate posterior $q(\mathbf{z} \mid \mathbf{q})$ produced by the hand-specific encoders, the latent loss is

$$L_3 = \mathcal{L}_{\text{KL}} = \mathbb{E}_{\mathbf{q}}\big[\text{KL}\big(q(\mathbf{z} \mid \mathbf{q}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big)\big], \tag{3}$$

which encourages the shared DexHand latent space to follow a $\mathcal{N}(0, I)$ distribution and facilitates sampling and interpolation across embodiments.

Training Data and Protocol. The latent autoencoder is trained *without* any demonstration or IK-generated trajectories. Instead, for each hand $s \in \mathcal{H}$ we randomly sample joint configurations within the hardware joint limits to form synthetic joint-position vectors $\mathbf{q}^{(s)}$. For every such sample we encode $\mathbf{q}^{(s)}$ to a latent $\mathbf{z}$, decode it through all decoders $\{D_t\}_{t\in\mathcal{H}}$, and accumulate reconstruction and retargeting losses: the self-decoding $D_s(\mathbf{z})$ contributes to L_1, while cross-hand decodings $D_t(\mathbf{z})$ for $t \neq s$ contribute to L_2. Losses from all hands are aggregated and a single backward pass is applied, so all encoders and decoders are optimized jointly. Because L_2 uses only forward kinematics of each hand and decoded poses, the alignment of the latent space across embodiments is completely self-supervised and does not require any paired cross-hand trajectories.

Total Latent Objective. The full latent training loss combines reconstruction, retargeting, and KL regularization:

$$L_{\text{latent}} = L_1 + L_2 + \beta L_3. \tag{4}$$

In all experiments we fix the weights to $\beta = 10^{-5}$, $\lambda_{\text{dis}} = 2000.0$, $\lambda_{\text{dir}} = 5.0$, and $\lambda_{\text{dis}}^{\text{exp}} = 12.0$. These values yield a latent space that is both geometrically well-aligned across hands and smooth enough to support sampling and interpolation.

4. Experiments

Tasks & Dataset. We design 10 diverse tasks with different skills and objects to evaluate our VLA models. For each task, we collect 50 demonstrations each task per hand via [18], with 2000 demonstrations collected in total. Task descriptions are listed below:

(a) Prepare Fruits (PF). Put the banana and orange on the green board for cutting.

(b) Stack Cans (SC). Stack the cheese can on top of the salt.

(c) Sort Cans (SoC). Put the tomato can and the cheese can into the container.

(d) Hand over Bottle (HB). Hand over the white bottle from right hand to left hand.

(e) Re-organize Lemons (RL). Put the yellow lemon and the

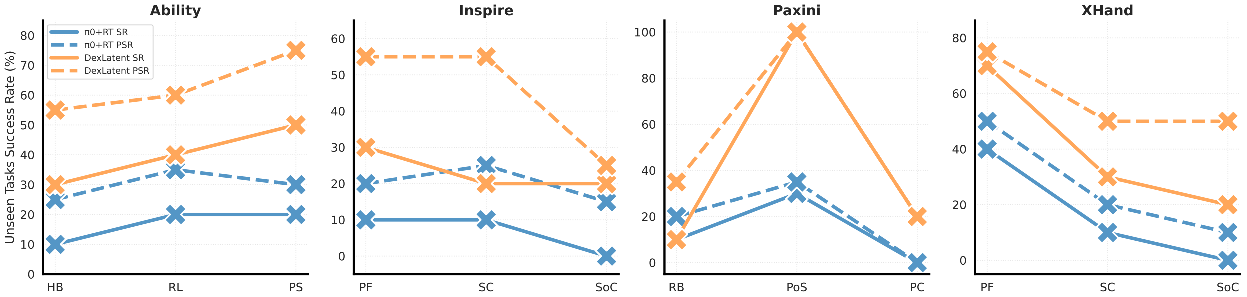


Figure 4. **Zero-shot Unseen Tasks Generalization**. For each hand, we randomly select some tasks as unseen tasks, whose data are held out from the training dataset. Then we test the unseen tasks with model trained on other data. Results show that by training with an aligned latent action space, XL-VLA gets the ability to generalize to novel hand-task combination in a zero-shot manner. PSR stands for "Partial Success Rate", where policy is rewarded with half success if only one arm finishes its task.

green lime into the bowl.

(f) Pour Sauce (PS). Pour mustard sauce into the meat can.

(g) Re-arrange Boxes (RB). Keep the table organized by re-arranging the two boxes.

(h) Push Sugar (PuS). Push the sugar boxes together.

(i) Pour Sugar (PoS). Add sugar to the starfruit.

(j) Push Cans (PC). Push the two tomato cans together.

Hardware. To evaluate our method, we conduct comprehensive experiments on our real-world robot platform. We use a bimanual 7-DoF xArm and a Unitree G1 humanoid with various robot hands, shown in table 3.

	Ability	Inspire	X-Hand1	Paxini DexH13
#Fingers	5	5	5	4
#DoF(mimic)	12(6)	12(6)	12	16(3)

Table 3. **Dexterous Hand Comparison.**

Experiment Settings. We initialize the XL-VLA with weights from [6], then train the model on our collected multi-embodiment dataset. We use 8 NVIDIA H100 GPUs to train XL-VLA, each having 80GB memory. The model is trained 60K steps with a batch size of 128. Note that XL-VLA is a unified cross-embodiment multi-task policy. We use language to condition the policy on multiple tasks.

4.1. Main Results

In this section, we are trying to answer the following questions that mainly focus on *Effectiveness of VLA + Latent Integration:* (1) Does XL-VLA outperform standard VLA models in cross-embodiment training? (2) Does XL-VLA enable zero-shot cross-embodiment skill transfer?

Cross-Hand Data Scaling. Tab.2 presents the cross-embodiment manipulation results for XL-VLA compared with a strong π_0 baseline trained on the same multi-hand, multi-task dataset as a single shared policy across all four hands—Ability, Inspire, Paxini, and X-Hand—and ten manipulation tasks. Although π_0 can nominally accommodate different embodiments by varying sequence lengths, its performance remains inconsistent and generally low due to substantial kinematic and actuation differences across hands. In contrast, XL-VLA achieves strong and consistent improvements for every hand and task, demonstrating the benefit of learning a shared latent action representation.

Across hands, XL-VLA yields notable per-embodiment gains. The Ability Hand, which features relatively simple actuation, benefits from a large boost in reliability, improving from 0.37 to 0.73 overall. The Paxini Hand achieves highest performance among all embodiments (0.78 overall) , indicating strong compatibility between its actuation structure and the learned latent mapping. XHand , which is the most mechanically distinct from the rest, also improves significantly from 0.29 to 0.70, showing that XL-VLA can bridge large embodiment gaps.

Averaged over all tasks and hands, XL-VLA increases the mean success rate from 0.55 to 0.90 (+0.35). Particularly large improvements are observed for dexterity-heavy tasks such as Sort Cans, Hand over Bottle, and Re-arrange Boxes, underscoring the effectiveness of our embodiment-invariant latent space in capturing fine-grained manipulation behavior. Overall, these results demonstrate that XL-VLA enables robust cross-embodiment action prediction and consistently surpasses VLA models that lack a unified action representation.

Cross-Robot Data Scaling. To show the unified latent space benefit even for different robot systems, we test four manipulation tasks with data from the tabletop xArm and humanoid G1. We co-train the data from all embodiments on the four tasks with the same training parameters and show the G1 success rates in figure 5. We can see that simply using aligned latent action space boost the performance of training on the raw action space, which has varied state/action lengths.

Zero-Shot Task Generalization. A key advantage of using an embodiment-invariant latent action space is its ability to support seamless *zero-shot generalization* to unseen tasks. Because all dexterous hands share the

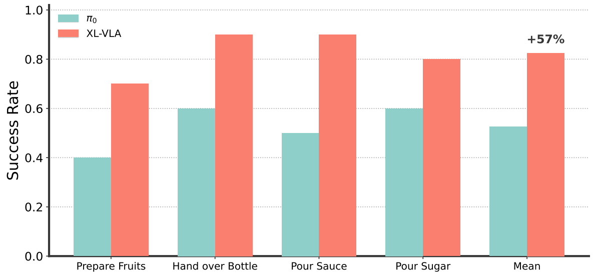


Figure 5. **G1 Cross-Robot Performance.** Co-training with latent xArm and humanoid data outperforms using raw actions.

same latent representation, a policy trained on a subset of tasks with one embodiment can transfer to a different task–hand combination without requiring additional training or retargeting.

To evaluate this capability, we hold out several manipulation tasks as *unseen tasks* for each hand and train XL-VLA on the remaining tasks. At test time, the trained policy is applied directly to the unseen task through the corresponding embodiment-specific decoder. As shown in Fig. 4, we report both *absolute success rate* (SR) and *partial success rate* (PSR), where PSR accounts for intermediate progress (0.25, 0.5, 0.75, 1.0) to provide a more fine-grained measure of policy performance.

For comparison, we construct a π_0+RT baseline in which a policy is trained on all tasks using only the XHand embodiment. During evaluation, we apply a standard *kinematic retargeting* algorithm to map the predicted XHand joint trajectories to the other embodiments (Inspire, Ability, Paxini) by aligning fingertip positions. This baseline reflects common practice in cross-embodiment manipulation and allows us to assess whether our latent action representation provides genuine zero-shot benefits over retargeting-based transfer.

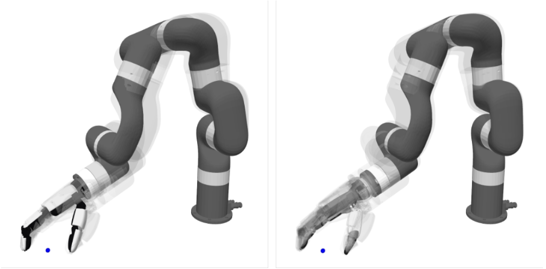
Across all embodiments and tasks, XL-VLA consistently outperforms the retargeting VLA baseline, often by a substantial margin. Notably, XL-VLA *never* underperforms the baseline on any hand or task, highlighting the robustness of the latent action representation. The gains are especially pronounced on fine-grained dexterous tasks (e.g., `HB`, `RB`), where geometric retargeting struggles to maintain coordinated finger motion.

4.2. Ablation Results

In this section, we are trying to answer the question about *Effectiveness of the Latent Action Space:* (1) How well does the learned latent space function as a retargeting mechanism on its own? (2) What is the impact of different design choices within the latent space, as shown through ablation studies?

Latent Replay Comparison. We further compare our latent action space against LAD [2], a supervised latent-space retargeting method. To ensure a fair and challenging evaluation, we perform *latent replay* by taking demonstrations from two embodiments and replaying them on the other two embodiments using each method's latent mapping. As shown in Table 4, our approach achieves a mean success rate of **0.82** and **0.81** on the two hand pairs (Ability+Inspire and Paxini+XHand), substantially outperforming LAD, which attains only **0.60** and **0.61**. This improvement is consistent across all tasks, with gains particularly pronounced on fine-grained manipulation tasks such as *SC*, *SoC*, and *HB*, where LAD exhibits noticeable degradation. Notably, our method achieves these results without any supervision data or paired labels, relying solely on unsupervised latent alignment. These findings highlight that our latent space captures embodiment-invariant structure more effectively than supervised alternatives, enabling significantly more reliable cross-hand trajectory replay.

Visual Result. Figure 6 shows latent decoding across different dexterous hands. We visualize one hand at full opacity and others with partial transparency, with the target grasp point marked in blue. Despite differing kinematics, all hands produce consistent poses from the same latent code, indicating that the learned latent space captures embodiment-invariant control.



(a) X-Hand (b) Inspire Hand

Figure 6. **Latent Visualizations.** Latent decoding results cross embodiment.

Design Choice Comparison. We conduct a comprehensive ablation study to evaluate architectural and loss-design choices for the latent action space, summarized in Tab. 5. Our final configuration uses the $H_{64}^{128\to64}$ architecture with a latent dimension of 32. All metrics follow a "lower is better" convention, and the worst result compared with our method within each row are highlighted in green. Across reconstruction, cross-embodiment retargeting, latent continuity, and interpolation smoothness, our design choice achieves relatively stronger performance. Notably, performance remains stable across a wide range of architectures and latent dimensions, with degradation only occurring when the latent size is significantly increased (e.g., L_{128}), suggesting that excessively large latent spaces hinder embodiment-invariant structure. These results indicate that our chosen con-

Model	Combination	PF	SC	SoC	HB	RL	PS	RB	PuS	PoS	PC	Mean
LAD [3]	Ability+Inspire	0.8	0.4	0.5	0.4	0.6	0.6	0.6	0.5	0.7	0.9	0.60
	Paxini+XHand	0.7	0.5	0.6	0.7	0.5	0.8	0.7	0.4	0.6	0.6	0.61
XL-VLA	Ability+Inspire	**0.9**	**0.7**	0.8	0.7	**0.8**	0.7	0.8	**0.8**	**1.0**	**1.0**	**0.82**
	Paxini+XHand	0.8	**0.7**	**0.9**	**0.8**	0.6	**0.9**	**0.9**	0.6	**1.0**	0.9	0.81

Table 4. **Latent replay comparison**. We compare our latent space with Latent Action Diffusion (LAD) [3]. For each hand combination, teleoperated trajectories collected on one source hand are encoded into the latent space, decoded onto the target hand, and replayed on real hardware. A replay is counted as successful if the encoded–decoded sequence can be executed without breaking contact or causing self-collisions. Higher replay success indicates better cross-embodiment consistency of the latent representation.

	Reconstruction ↓		**Cross Embodiment ↓**				**Latent Continuity ↓**		**Interpolation ↓**	
Exp	**Joint**	**Tip**	**PT$^{\text{dir}}$**	**PT$^{\text{dist}}$**	**RT$^{\text{dir}}$**	**RT$^{\text{dist}}$**	**Joint**	**Tip**	**Accel.**	**Jerk**
Ours	5.476	3.703	11.857	1.872	10.492	6.295	4.492	8.534	8.683	9.659
$-L_1$	61.672	39.400	11.741	1.857	10.398	6.375	24.784	58.858	12.028	16.852
$-L_2^{\text{dist}}$	5.195	3.580	3.972	4.413	6.788	24.488	4.073	8.168	8.525	9.240
$-L_2^{\text{dir}}$	4.966	3.378	46.217	2.251	53.546	5.518	4.451	9.217	8.742	9.551
$-L_2$ (both)	3.781	2.602	62.733	8.080	71.765	62.809	2.823	6.757	8.602	9.426
H_{256}^{128}	5.897	3.908	9.073	1.613	10.432	6.277	3.104	6.410	9.213	10.406
$H_{128}^{64} \times 2$	8.216	4.280	9.027	1.513	10.572	6.713	5.004	8.832	8.559	9.479
H_{64}^{64}	4.979	3.411	9.010	1.655	10.702	6.985	2.922	6.298	8.618	9.296
H^{64}	5.021	3.445	9.010	1.518	10.213	6.435	4.174	8.132	8.246	8.740
$\mathbf{L}_8$	20.913	6.499	9.217	1.557	10.960	6.805	8.164	11.720	8.758	9.778
$\mathbf{L}_{16}$	8.416	4.159	13.624	1.989	11.084	6.558	5.445	9.192	8.436	8.996
$\mathbf{L}_{64}$	5.542	3.583	8.314	1.549	10.995	6.955	4.140	8.174	8.299	8.944
$\mathbf{L}_{96}$	5.239	3.422	9.332	1.562	10.516	6.554	3.498	7.072	8.700	9.703
$\mathbf{L}_{128}$	5.324	3.543	8.736	1.529	10.286	6.215	3.282	6.882	8.607	9.294

Table 5. **Ablations.** Ablation results comparing reconstruction accuracy, cross-embodiment retargeting, latent-space continuity, and interpolation smoothness. **Exp** denotes model variants: removing losses ($-L_2$, $-L_2^{dist}$, $-L_2^{dir}$, or both), changing hidden sizes (H_a^b), or changing latent dimension (L_d). Metrics include joint and tip RMSE for reconstruction; pinch- and random-motion direction/distance errors for retargeting; joint/tip latent continuity; and mean acceleration/jerk for interpolation.

figuration offers an effective balance between model capacity and latent compactness.

And we explicitly write the evaluation metrics we design for this ablation study below:

Recon Joint/Tip RMSE. Evaluates reconstruction fidelity of one hand. Synthetic random joint configurations are encoded and decoded, and we report the root-mean-square error (RMSE) between input and reconstructed joint angles (radians). In parallel, the original and reconstructed configurations are passed through each hand's forward-kinematics model to obtain fingertip positions, and we compute the RMSE of fingertip displacement (meters). Lower values indicate that the latent representation preserves hand configurations.

Pinch and Random Tip Dir./Dist. Error Assesses cross-hand transfer for pinch and random grasps. Pinch poses and random poses are encoded on a source hand and decoded on each target hand; for each pose, we form a line from the thumb tip to the opposing fingertip. Directional error is measured as the angle between predicted and reference lines (degrees), and distance error is measured as the absolute difference in thumb–finger distance (meters). Smaller values for both components indicate less directional drift and more faithful preservation of pinch aperture across hands.

Latent Continuity (Joint/Tip) Test the local smoothness of the latent manifold. Encoded hand latents are perturbed with isotropic Gaussian noise of standard deviation $\epsilon = 0.05$ and decoded back to joint angles, from which we compute the norm of the resulting joint-space deviation (radians). The corresponding finger tip effect is obtained by forwarding both perturbed and unperturbed reconstructions through forward kinematics and measuring the norm of fingertip displacement (meters). Small deviations indicate that the latent representation varies smoothly.

Interp. Accel./Jerk Mean Characterizes the smoothness of latent-space interpolations. Two poses are encoded, and their latent codes are linearly interpolated. Decoding these intermediate codes yields fingertip trajectories, from which finite differences provide velocities and accelerations. Interp. Accel. Mean is the mean acceleration norm (meters per normalized interpolation step squared), while Interp. Jerk Mean is the mean norm of the jerk (the finite-difference derivative of acceleration, meters per step cubed). Lower values for both indicate smoother interpolation paths in the latent space.

5. Conclusion.

In this work, we introduced XL-VLA, a vision–language–action framework equipped with a unified latent action space for scalable cross-embodiment dexterous manipulation. By learning an embodiment-invariant latent representation, our approach enables seamless training across diverse robotic hands and supports zero-shot generalization to new hand–task combinations. Extensive real-world experiments demonstrate that XL-VLA consistently outperforms standard VLA models and retargeting-based baselines, while offering a flexible and plug-and-play interface for newly introduced hands. Overall, our results highlight latent action spaces as a powerful foundation for building generalizable, data-efficient dexterous manipulation systems. We believe this work takes a step toward more unified and adaptable robotic manipulation frameworks capable of keeping pace with rapid hardware innovation.

References

[1] Yang Bai, Liudi Yang, George Eskandar, Fengyi Shen, Dong Chen, Mohammad Altillawi, Ziyuan Liu, and Gitta Kutyniok. Roboswap: A gan-driven video diffusion framework for unsupervised robot arm swapping. *arXiv preprint arXiv:2506.08632*, 2025. 2

[2] Erik Bauer, Elvis Nava, and Robert K. Katzschmann. Latent action diffusion for cross-embodiment manipulation. *arXiv preprint arXiv:2506.14608*, 2025. 7

[3] Erik Bauer, Elvis Nava, and Robert K. Katzschmann. Latent action diffusion for cross-embodiment manipulation. *arXiv preprint arXiv:2506.14608*, 2025. 2, 3, 8

[4] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025. 14, 15

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 3

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. 3, 4, 5, 6, 15

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 3

[8] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, 2025. 2, 3

[9] Qingwen Bu, Yanting Yang, Jisong Yang, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 2

[10] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017. 14

[11] Zhefeng Cao, Ben Liu, Sen Li, Wei Zhang, and Hua Chen. G-dream: Graph-conditioned diffusion retargeting across multiple embodiments. *arXiv preprint arXiv:2505.20857*, 2025. 2

[12] Annie S Chen, Philemon Brakel, Antonia Bronars, Annie Xie, Sandy Huang, Oliver Groth, Maria Bauza, Markus Wulfmeier, Nicolas Heess, and Dushyant Rao. Exploiting policy idling for dexterous manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. 2

[13] Xu Chi, Chao Zhang, Yang Su, Lingfeng Dou, Fujia Yang, Jiakuo Zhao, Haoyu Zhou, Xiaoyou Jia, Yong Zhou, and Shan An. Open teledex: A hardware-agnostic teleoperation system for imitation learning based dexterous manipulation. *arXiv preprint arXiv:2510.14771*, 2025. 2

[14] Xu Chi, Chao Zhang, Yang Su, Lingfeng Dou, Fujia Yang, Jiakuo Zhao, Haoyu Zhou, Xiaoyou Jia, Yong Zhou, and Shan An. Open teledex: A hardware-agnostic teleoperation system for imitation learning based dexterous manipulation. *arXiv preprint arXiv:2510.14771*, 2025. 3

[15] Prithwish Dan, Kushal Kedia, Angela Chao, Edward Weiyi Duan, Maximus Adrian Pace, Wei-Chiu Ma, and Sanjiban Choudhury. X-sim: Cross-embodiment learning via real-to-sim-to-real. In *Proceedings of the 9th Conference on Robot Learning (CoRL 2025)*, pages 816–833. PMLR, 2025. 2

[16] Apan Dastider, Hao Fang, and Mingjie Lin. Cross-embodiment robotic manipulation synthesis via guided demonstrations through cyclevae and human behavior transformer. *arXiv preprint arXiv:2503.08622*, 2025. 2, 3

[17] Travis Davies, Yiqi Huang, Yunxin Liu, Xiang Chen, Huxian Liu, and Luhui Hu. Tenma: Robust cross-embodiment robot manipulation with diffusion transformer. *arXiv preprint arXiv:2509.11865*, 2025. 2, 3

[18] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. 2024. 5, 14

[19] Yunlong Dong, Xing Liu, Jun Wan, and Zelin Deng. Gex: Democratizing dexterity with fully-actuated dexterous hand and exoskeleton glove. *arXiv preprint arXiv:2506.04982*, 2025. 3

[20] Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, Cheng Zhang, Yufeng Zhu, Stephen Tian, Zhengqin Li, Geordie Moffatt, et al. Digital twin catalog: A large-scale photorealistic 3d object digital twin dataset. *arXiv preprint arXiv:2504.08541*, 2025. 14

[21] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of the 8th Conference on Robot Learning (CoRL 2024)*, pages 496–512. PMLR, 2025. 2

[22] Ainaz Eftekhar, Rose Hendrix, Luca Weihs, Jiafei Duan, Ege Caglar, Jordi Salvador, Alvaro Herrasti, Winson Han, Eli VanderBil, Aniruddha Kembhavi, Kiana Ehsani, Kuo-Hao Zeng, and Ranjay Krishna. The one ring: a robotic indoor navigation generalist. *arXiv preprint arXiv:2412.14401*, 2024. 2

[23] Hao-Shu Fang, Branden Romero, Yichen Xie, Arthur Hu, Bo-Ruei Huang, Juan Alvarez, Matthew Kim, Gabriel Margolis, Kavya Anbarasu, Masayoshi Tomizuka, et al. Dexop: A device for robotic transfer of dexterous human manipulation. In *Workshop on Dexterous Manipulation at Robotics: Science and Systems (RSS)*, 2025. Workshop paper. 2

[24] Xin Fei, Zhixuan Xu, Huaicong Fang, Tianrui Zhang, and Lin Shao. T(r,o) grasp: Efficient graph diffusion of robot-object spatial transformation for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2510.12724*, 2025. 2

[25] Mateo Guaman Castro, Sidharth Rajagopal, Daniel Gorbatov, Matt Schmittle, Rohan Baijal, Octi Zhang, Rosario Scalise, Sidharth Talia, Emma Romig, Celso de Melo, Byron Boots, and Abhishek Gupta. Vamos: A hierarchical vision-language-action model for capability-modulated and steerable navigation. *arXiv preprint arXiv:2510.20818*, 2025. 2

[26] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2

[27] Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *European Conference on Computer Vision*, pages 478–495. Springer, 2024. 2

[28] Adam Hung, Fan Yang, Abhinav Kumar, Sergio Aguilera Marinovic, Soshi Iba, Rana Soltani Zarrin, and Dmitry Berenson. Avo: Amortized value optimization for contact mode switching in multi-finger manipulation. *arXiv preprint arXiv:2510.07548*, 2025. 2

[29] Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pretrain robots: Manipulation-centric robotic representation from large-scale robot datasets. *arXiv preprint arXiv:2410.22325*, 2024. 3

[30] Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025. 2

[31] Changwei Jing, Jai Krishna Bandi, Jianglong Ye, Yan Duan, Pieter Abbeel, Xiaolong Wang, and Sha Yi. Contact-aware neural dynamics. *arXiv preprint arXiv:2601.12796*, 2026. 2

[32] Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. In *Proceedings of the 9th Conference on Robot Learning (CoRL 2025)*. PMLR, 2025. 2

[33] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 3

[34] Yuki Kuroda, Tomoya Takahashi, Cristian C Beltran-Hernandez, Masashi Hamaya, and Kazutoshi Tanaka. Plexus hand: Lightweight four-motor prosthetic hand enabling precision-lateral dexterous manipulation. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2025. 2

[35] Arjun S. Lakshmipathy, Jessica K. Hodgins, and Nancy S. Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations. *arXiv preprint arXiv:2402.04820*, 2024. 3

[36] Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025. 2

[37] Xiaoyi Lin, Kunpeng Yao, Lixin Xu, Xueqiang Wang, Xuetao Li, Yuchen Wang, and Miao Li. Dexflow: A unified approach for dexterous hand pose retargeting and interaction. *arXiv preprint arXiv:2505.01083*, 2025. 3

[38] Yuhao Lin, Yi-Lin Wei, Haoran Liao, Mu Lin, Chengyi Xing, Hao Li, Dandan Zhang, Mark Cutkosky, and Wei-Shi Zheng. Typetele: Releasing dexterity in teleoperation by dexterous manipulation types. *arXiv preprint arXiv:2507.01857*, 2025. 3

[39] Jiacheng Liu, Pengxiang Ding, Qihang Zhou, Yuxuan Wu, Da Huang, Zimian Peng, Wei Xiao, Weinan Zhang, Lixin Yang, Cewu Lu, and Donglin Wang. Trajbooster: Boosting humanoid whole-body manipulation via trajectory-centric learning. *arXiv preprint arXiv:2509.11839*, 2025. 2

[40] Xueyi Liu, He Wang, and Li Yi. Dexndm: Closing the reality gap for dexterous in-hand rotation via joint-wise neural dynamics model. *arXiv preprint arXiv:2510.08556*, 2025. 2

[41] Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025. 3

[42] Yaru Niu, Yunzhe Zhang, Mingyang Yu, Changyi Lin, Chenhao Li, Yikai Wang, Yuxiang Yang, Wenhao Yu, Tingnan Zhang, Zhenzhen Li, Jonathan Francis, Bingqing Chen, Jie Tan, and Ding Zhao. Human2locoman: Learning versatile quadrupedal manipulation with human pretraining. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 2

[43] Guoping Pan, Qingwei Ben, Zhecheng Yuan, Guangqi Jiang, Yandong Ji, Jiangmiao Pang, Houde Liu, and Huazhe Xu. Roboduet: A framework affording mobile-manipulation and cross-embodiment. *arXiv preprint arXiv:2403.17367*, 6, 2024. 2

[44] En Yen Puang, Federico Ceola, Giulia Pasquale, and Lorenzo Natale. Pchands: Pca-based hand pose synergy representation on manipulators with n-dof. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2025. 2

[45] Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y. Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Poster. 2, 3

[46] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025. 2

[47] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 2

[48] Benjamin A Richardson, Felix Grüninger, Lukas Mack, Joerg Stueckler, and Katherine J Kuchenbecker. Isyhand: A dexterous multi-finger robot hand with an articulated palm. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Seoul, Korea, 2025. 2

[49] Mingyo Seo, H. Andy Park, Shenli Yuan, Yuke Zhu, and Luis Sentis. Legato: Cross-embodiment imitation using a grasping tool. *arXiv preprint arXiv:2411.03682*, 2024. 2

[50] Wenxuan Song, Jiayi Chen, Pengxiang Ding, Yuxin Huang, Han Zhao, Donglin Wang, and Haoang Li. Ceedvla: Consistency vision-language-action model with early-exit decoding. *arXiv preprint arXiv:2506.13725*, 2025. 2

[51] Kit-Wa Sou, Junhao Gong, Shoujie Li, Chuqiao Lyu, Ziwu Song, Shilong Mu, and Wenbo Ding. Moirétac: A dual-mode visuotactile sensor for multidimensional perception using moiré pattern amplification. *arXiv preprint arXiv:2509.12714*, 2025. 2

[52] Wentao Tan, Bowen Wang, Heng Zhi, Chenyu Liu, Zhe Li, Jian Liu, Zengrong Lin, Yukun Dai, Yipeng Chen, Wenjie Yang, Enci Xie, Hao Xue, Baixu Ji, Chen Xu, Zhibin Wang, Tianshi Wang, Lei Zhu, and Heng Tao Shen. Blm$_1$: A boundless large model for cross-space, cross-task, and cross-embodiment learning. *arXiv preprint arXiv:2510.24161*, 2025. 2

[53] Tianyu Wang, Dwait Bhatt, Xiaolong Wang, and Nikolay Atanasov. Cross-embodiment robot manipulation skill transfer using latent space alignment. *arXiv preprint arXiv:2406.01968*, 2024. 2, 3

[54] Tianyu Wang, Dwait Bhatt, Xiaolong Wang, and Nikolay Atanasov. Cross-embodiment robot manipulation skill transfer using latent space alignment. *arXiv preprint arXiv:2406.01968*, 2024. 2

[55] Yi-Lin Wei, Zhexi Luo, Yuhao Lin, Mu Lin, Zhizhao Liang, Shuoyu Chen, and Wei-Shi Zheng. Omnidexgrasp: Generalizable dexterous grasping via foundation model and force feedback. *arXiv preprint arXiv:2510.23119*, 2025. 2

[56] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024. 3

[57] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. In *Proceedings of the 9th Conference on Robot Learning (CoRL 2025)*. PMLR, 2025. 2

[58] Ruoshi Wen, Jiajun Zhang, Guangzeng Chen, Zhongren Cui, Min Du, Yang Gou, Zhigang Han, Junkai Hu, Liqun Huang, Hao Niu, Wei Xu, Haoxiang Zhang, Zhengming Zhu, Hang Li, and Zeyu Ren. Dexterous teleoperation of 20-dof bytedexter hand via human motion retargeting. *arXiv preprint arXiv:2507.03227*, 2025. 3

[59] Zhiyuan Wu, Rolandos Alexandros Potamias, Xuyang Zhang, Zhongqun Zhang, Jiankang Deng, and Shan Luo. Cedex: Cross-embodiment dexterous grasp generation at scale from human-like contact representations. *arXiv preprint arXiv:2509.24661*, 2025. 2

[60] Chendong Xin, Mingrui Yu, Yongpeng Jiang, Zhefeng Zhang, and Xiang Li. Analyzing key objectives in human-to-robot retargeting for dexterous manipulation. *arXiv preprint arXiv:2506.09384*, 2025. 3

[61] Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, pages 437–459, 2025. 2

[62] Sirui Xu, Yu-Wei Chao, Liuyu Bian, Arsalan Mousavian, Yu-Xiong Wang, Liang-Yan Gui, and Wei Yang. Dexplore: Scalable neural control for dexterous manipulation from reference-scoped exploration. *arXiv preprint arXiv:2509.09671*, 2025. 3

[63] Zhuowei Xu, Zilin Si, Kevin Zhang, Oliver Kroemer, and Zeynep Temel. A multi-modal tactile fingertip design for robotic hands to enhance dexterous manipulation. *arXiv preprint arXiv:2510.05382*, 2025. 2

[64] Rui Yan, Jiajian Fu, Shiqi Yang, Lars Paulsen, Xuxin Cheng, and Xiaolong Wang. Ace-f: A cross embodiment foldable system with force feedback for dexterous teleoperation, 2025. 14

[65] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025. 2, 3

[66] Shunpeng Yang, Zhen Fu, Zhefeng Cao, Guo Junde, Patrick Wensing, Wei Zhang, and Hua Chen. Multi-loco: Unifying multi-embodiment legged locomotion via reinforcement learning augmented diffusion. In *Proceedings of the 9th Conference on Robot Learning (CoRL 2025)*, pages 1030–1048. PMLR, 2025. 2

[67] Jianglong Ye, Lai Wei, Guangqi Jiang, Changwei Jing, Xueyan Zou, and Xiaolong Wang. From power to precision: Learning fine-grained dexterity for multi-fingered robotic hands. *arXiv preprint arXiv:2511.13710*, 2025. 2

[68] Zhao-Heng Yin, Changhao Wang, Luis Pineda, Krishna Bodduluri, Tingfan Wu, Pieter Abbeel, and Mustafa Mukadam. Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm. *arXiv preprint arXiv:2503.07541*, 2025. 3

[69] Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, et al. Forcevla: Enhancing vla models with a force-aware moe for contact-rich manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Poster. 2

[70] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning (CoRL 2021)*, pages 537–546. PMLR, 2021. 2

[71] Yang Zhang, Chenwei Wang, Ouyang Lu, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, and Xuelong Li. Align-then-steer: Adapting the vision-language action models through unified latent guidance. *arXiv preprint arXiv:2509.02055*, 2025. 2, 3

[72] 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. 3

[73] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. *arXiv preprint arXiv:2501.10105*, 2025. 2

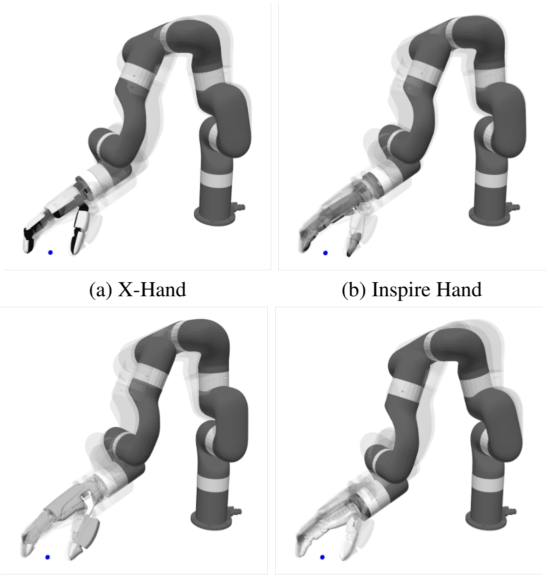
[74] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025. 2

[75] Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025. 2

6. Appendix

6.1. Latent Visualizations

In addition to Figure 5, we include visualizations of additional hands in Figure 7. This figure illustrates how the same latent representation is decoded across all four hands featured in our main paper. Furthermore, Figure 17 presents a continuous trajectory rendered for all hands, with the X-Hand highlighted for clarity.


(a) X-Hand

(b) Inspire Hand

(c) Paxini Hand

(d) Ability Hand

Figure 7. **More Latent Visualizations.** Latent decoding results cross embodiment.

6.2. Hardware Setup

Tabletop Scene Description. For the real-world experiments, we use a bimanual arm with tabletop settings. The arms are mounted on the edge of the table. The distance between the arms are 80cm. Each hand is connected to the end effector of the arm with 3D-printed mounts. Figure 8 shows the real-world tabletop scene with a pair of XHand, and figure 9 includes all the dexterous hands we use in our experiments.

Camera Description. We use a single RealSense L515 camera (round-shaped) mounted in front of the bimanual arms as the input view of the policy training. The camera pose is shown in figure 8 and the camera view is in figure 10. The raw resolution of RGB recordings from the camera is 960 × 540.

Humanoid Scene Description. Similar to xArm, we let G1 stand in front of a table. We use the same camera setting and mount L515 on the chest of G1 to have an egocentric view. Consider the mechanic design of G1,

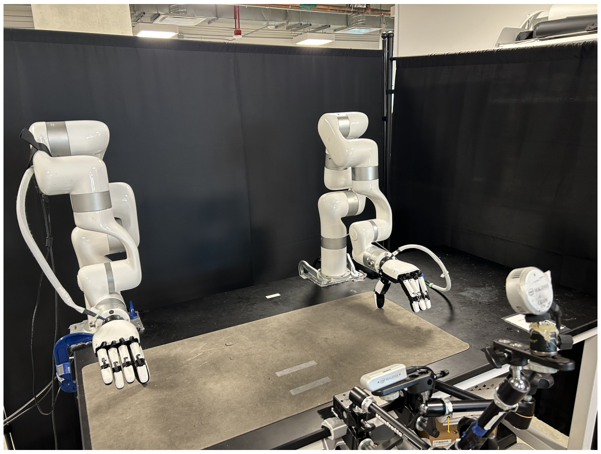


Figure 8. **xArm Camera Setup.** We use a single RealSense L515 camera with the front view. Note that the D435 camera here is not used for XL-VLA.

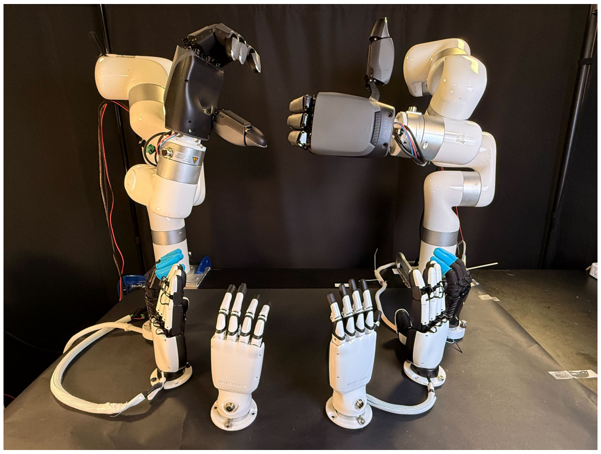


Figure 9. **Dexterous Hands.** We use 4 kinds of hands, with various shapes, scales, degrees of freedom, and actuated joints.

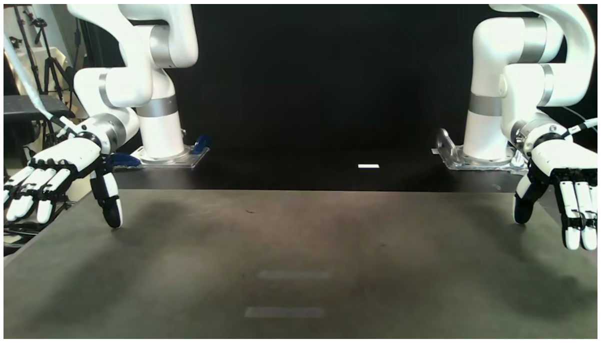


Figure 10. **xArm Camera View.** This is what our camera sees and also the input for XL-VLA and all the baseline methods.

we only use the Inspire hand since it is light. Figure 11 and 12 show the real-world G1 setting.

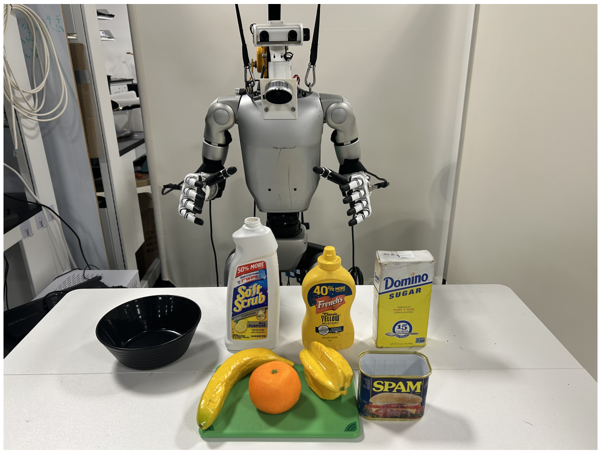


Figure 11. **G1 Scene.** We mount an L515 camera near the neck of G1 to have an egocentric view.

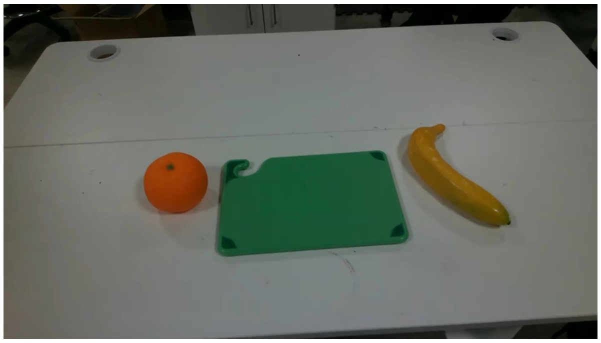


Figure 12. **G1 Egocentric Camera View.**

Object Description. To demonstrate that XL-VLA is capable of doing various manipulation tasks, we use diverse objects in our experiments, most of which are common everyday objects from existing datasets [10, 20]. The objects vary in scale, shape, texture, weight, etc. All the objects used in listed in figure 13.

6.3. Policy Learning Details

xArm Data Collection. We use Apple Vision Pro as the data collection tool [18], shown in figure 14. During data collection, our teleoperator wears the VR headset and get the tracked hand poses and wrist poses from the headset. We use these data to do robot hand retargeting and Inverse Kinematics.

Unitree G1 Data Collection. G1 is standing in front of a table, and we adapt HOMIE [4] and ACE-F [64] to do the upper-body teleoperation. We only use the upper-body system from HOMIE and replace their glove with the MANUS Mocap glove.

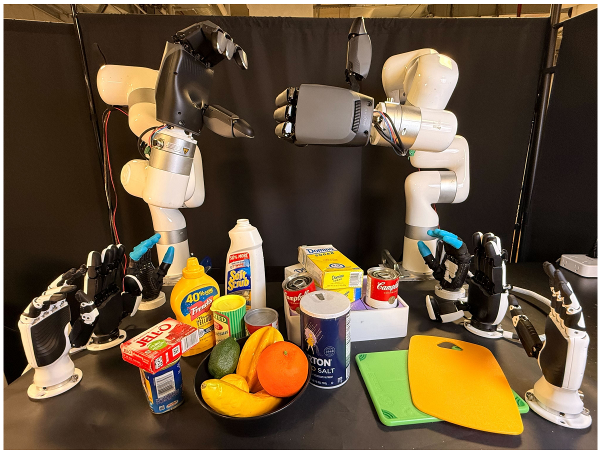


Figure 13. **Objects.** We use various everyday objects from existing datasets. They vary in scale, shape, texture, weight, etc., thus requiring the manipulation policy to be robust.

Figure 14. **Apple Vision Por for Data Collection.** Apple Vision Pro is used to track the human teleoperator's hands and wrists. Then the tracked data is processed with retargeting and Inverse Kinematics to control the real-world robots.

Task Visualizations. Using XHand as an example, we show the real-world task visualizations in figure 16. Four of the tasks are also tested on G1.

Model Training. Besides the model training details provided in the main paper, here are some more training details. RGB images is cropped and then resized from 960 × 540 to 320 × 240 during data post-processing. When loaded to train XL-VLA, they are resized to 224 × 224. We use natural language description as the task specification, which is part of the policy condition. The training usually takes around 10 hours for one multi-task policy.

Policy Evaluation. For the real-world evaluation, we do 10 trials for each experiment setting. Among these trials, object positions are randomly initialized while the

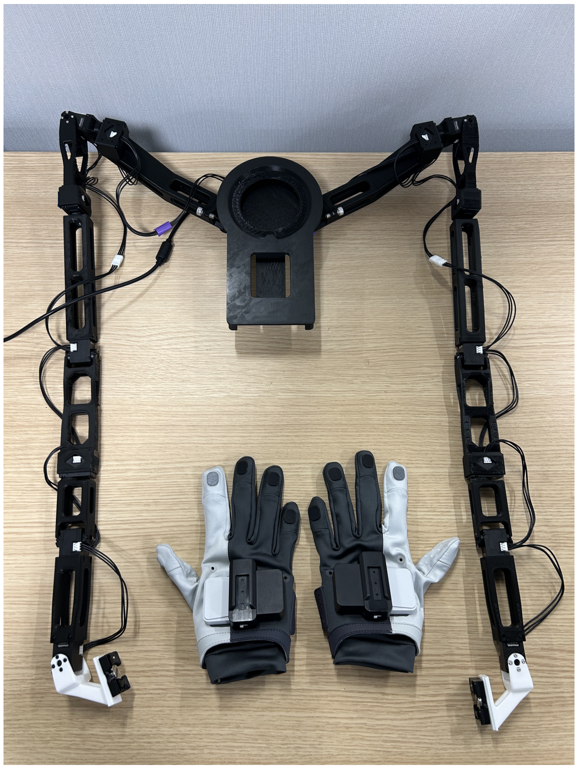


Figure 15. **G1 Teleoperation System.** We build the G1 upper-body teleoperation system from HOMIE [4]. We use a pair of MANUS Mocap glove to track the human hand pose.

initial joint position for the robot arm and hand remains the same for the same hand.

For unseen tasks only, we record the partial success rate (PSR). If any of the bimanual robot arm finishes its task and the whole task is failed, the overall success rate is 0.5. For other experiments, we do not use PSR. Only rollout that completes a specified task is count as a success.

6.4. G1 Experiment Results

Table 6 is the numeric results for figure 5.

Method	PF	HB	PS	PoS	Mean
π_0 [6]	0.4	0.6	0.5	0.6	0.525
XL-VLA	**0.7**	**0.9**	**0.9**	**0.8**	**0.825** +57%

Table 6. **G1 Policy Performances.**

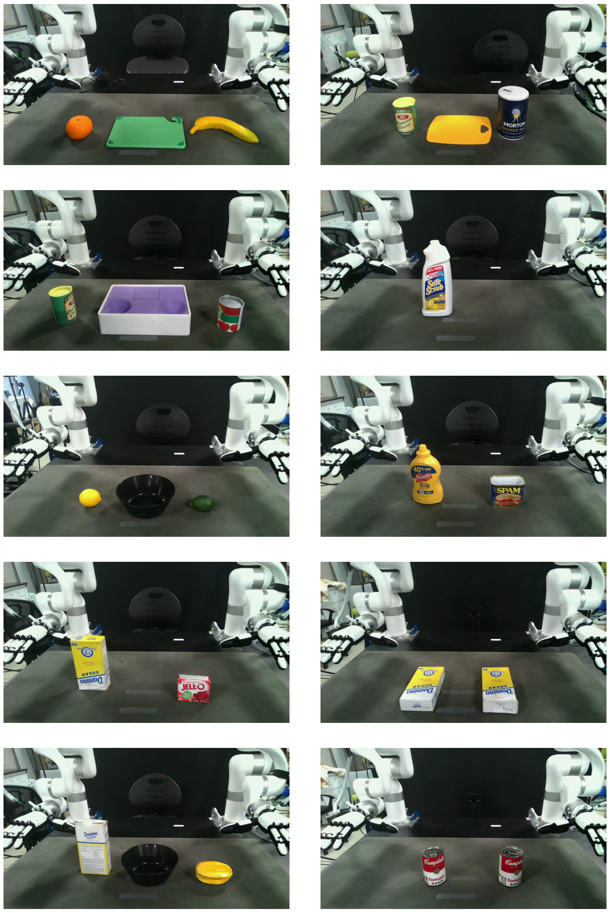


Figure 16. **Task Visualizations.** We design 10 various tasks to test all the models. The tasks require varied manipulation skills and have different difficulties.

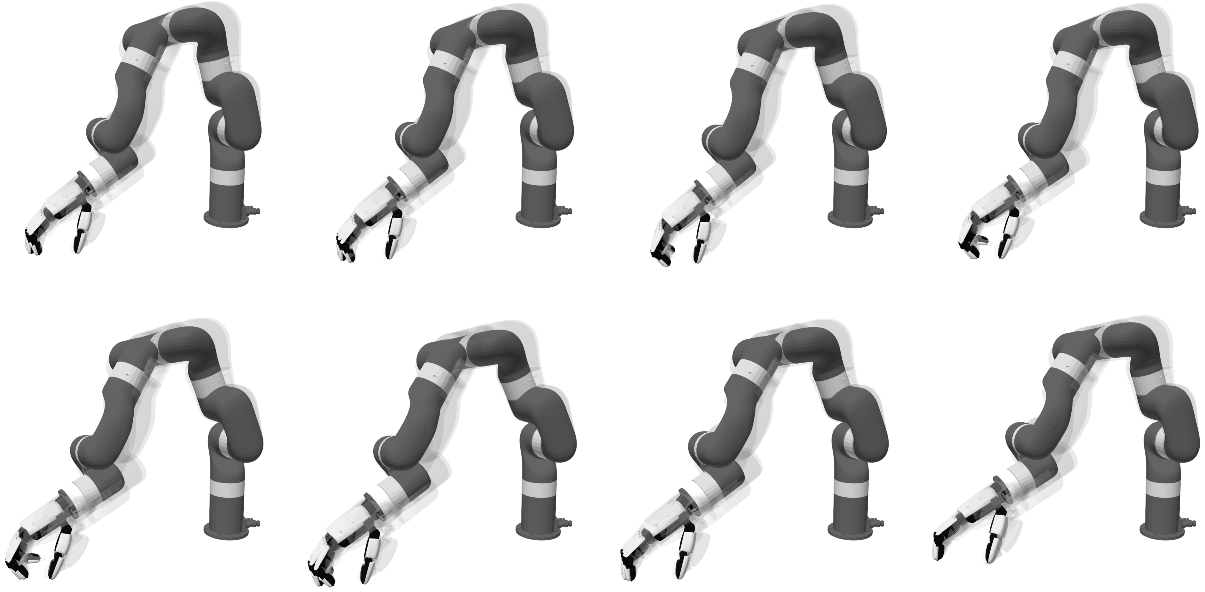


Figure 17. **Latent Visualization of a Grasping Trajectory.** A trajectory is shown here with all the robot hands.