villa-X: Enhancing Latent Action Modeling in Vision-Language-Action Models

Xiaoyu Chen[2*†], Hangxing Wei[3*†], Pushi Zhang[1*], Chuheng Zhang[1*], Kaixin Wang[1*], Yanjiang Guo[2], Rushuai Yang[4†], Yucen Wang[5†], Xinquan Xiao[2†], Li Zhao[1*‡], Jianyu Chen[2], and Jiang Bian[1]

[1]Microsoft Research, [2]Tsinghua University, [3]Wuhan University, [4]Hong Kong University of Science and Technology, [5]Nanjing University

Visual-Language-Action (VLA) models have emerged as a popular paradigm for learning robot manipulation policies that can follow language instructions and generalize to novel scenarios. Recent work has begun to explore the incorporation of *latent actions*, an abstract representation of visual change between two frames, into VLA pre-training. In this paper, we introduce villa-X, a novel Visual-Language-Latent-Action (ViLLA) framework that advances latent action modeling for learning generalizable robot manipulation policies. Our approach improves both *how latent actions are learned* and *how they are incorporated into VLA pre-training*. Together, these contributions enable villa-X to achieve superior performance across simulated environments including SIMPLER and LIBERO, as well as on two real-world robot setups including gripper and dexterous hand manipulation. We believe the ViLLA paradigm holds significant promise, and that our villa-X provides a strong foundation for future research.

Keywords: Latent Action, Vision-Language-Action Model

code: https://github.com/microsoft/villa-x
site: https://aka.ms/villa-x

1 Introduction

Latent action learning has emerged as a promising approach for the pretraining of vision-language-action (VLA) models [5, 11, 13, 30, 31, 35, 52, 68], enabling learning from both robot data and human video data [1, 11, 52, 67]. At the core of these methods is a Latent Action Model (LAM), which compresses visual changes between successive frames into latent tokens, referred to as latent actions. These latent actions are expected to capture motion semantics. These tokens function as pseudo-action labels for imitation learning and enriching robot policy training with abundant, action-free data.

Yet two questions remain: *how to learn latent actions* and *how to incorporate latent actions into VLA pre-training*? In this paper, we introduce villa-X, a novel Visual-Language-Latent-Action (ViLLA) framework that advances both key aspects of latent action modeling. For the latent action learning component, a key limitation is the absence of direct alignment between visual transitions and underlying robot states and actions. Although robot data with low-level states and actions are often available alongside action-free videos during pretraining, current methods choose to ignore these signals and focus solely on the visual part. As a result, latent actions can remain ungrounded in the robot's physical dynamics, resulting in a weak correspondence between latent action and robot behavior. To address this issue, we incorporate a proprio Forward Dynamics Model (FDM) module into the LAM as an auxiliary decoder. This module predicts future robot proprioceptive states and actions, encouraging the learned latent actions to better reflect the agent's behavior—not just the observed visual changes. Moreover, this design improves the interpretability of the learned latent actions and enables them to be more easily translated into executable robot actions. As a result, latent actions become a robust intermediary between vision-based representations and low-level controls. To exploit this structure, we propose to model the latent action and robot action distribution jointly through a joint diffusion process, and conditioning robot action generation on latent actions generation through attention. Compared to existing methods on latent action pre-training, our approach allows for more effective and structured information transfer from latent actions to robot actions.

We conduct comprehensive evaluations of villa-X on two simulated environments, SIMPLER and LIBERO, as well as two real-world setups, including various robot platforms with gripper and dexterous

*Equal contribution. †Interns at Microsoft Research. ‡Project lead.

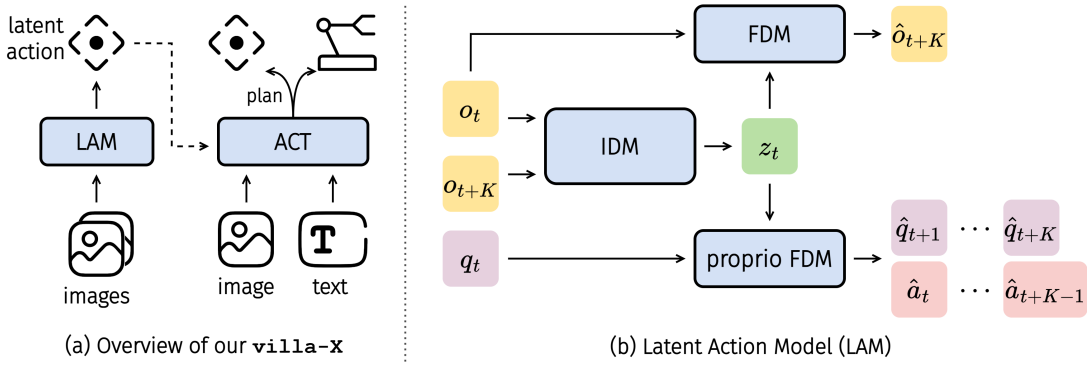


Figure 1: (a) High-level architecture of our villa-X method. (b) Detailed structure of the Latent Action Model (LAM) component within villa-X.

hand manipulation. Our results demonstrate that villa-X achieves superior performance, laying a robust foundation for future research in this domain.

Our contributions are listed as follows:

- We improve latent action learning by introducing an extra proprio FDM, which aligns latent tokens with underlying robot states and actions and grounds them in physical dynamics.
- We propose to jointly learn a latent action expert and a robot action expert through joint diffusion in the policy model, conditioning robot action prediction on latent actions to fully exploit their potential.
- Our method demonstrates superior performance on simulated environments as well as on real-world robotic tasks. The latent action expert can effectively plan into future with both visual and proprio state planning.

2 Related Work

Visual-Language-Action Models Vision-Language-Action (VLA) models [5, 11, 13, 30, 31, 35, 52, 68] leverage pre-trained vision-language models (VLMs) to generate robot actions using visual and language cues. They either directly repurpose VLMs for action prediction [11, 30, 35, 67] or use action experts to map VLM outputs to robot actions [5, 31, 52]. While training on large-scale datasets like Open X-Embodiment [13] enhance the generalization ability of VLAs, cross-embodiment generalization remains challenging due to diverse robot setups and configurations. Utilizing unlabeled trajectory data with pseudo-labels such as latent actions [11, 67], language sub-goals [49], or visual sub-goals [68] supports overcoming these challenges. Our method adopts the latent action approach and improves both the modeling of latent actions and their integration into VLA pretraining.

Modeling Latent Actions for VLA Pretraining Recent exploration into latent actions began with LAPO [58] and Genie [7], which primarily focused on the video game domain. Dynamo [15] adopted a similar architecture, using inverse and forward dynamics models to shape state representations. LAOM [51] propose to use supervision to learn better latent actions in the presense of distractors on Mujoco environments.

For robotic learning, methods have started to incorporate latent actions into VLA pretraining [1, 8, 10, 11, 38, 52, 66, 67]. LAPA [67] proposes to learn from videos, and trains its latent actions and Vision-Language Model (VLM) using either human or robot video data. Concurrently, IGOR [10] learns latent actions from a mixture of human and robot videos, marking the first demonstration of successful latent action transfer between humans and robots in a unified action space for embodied AI. Moto-GPT [11] co-fine-tunes both latent and robot action labels. GR00T [52] treats latent actions as a distinct embodiment, while Go-1 [1] generates robot actions conditioned on discrete latent tokens. UniVLA [8] proposes a two-stage training pipeline to learn task-centric latent actions. More recent works like [38, 66] explore the continuous latent actions. By contrast, our approach jointly models latent and robot actions through a joint diffusion process, conditioning robot action generation on latent actions for more effective and

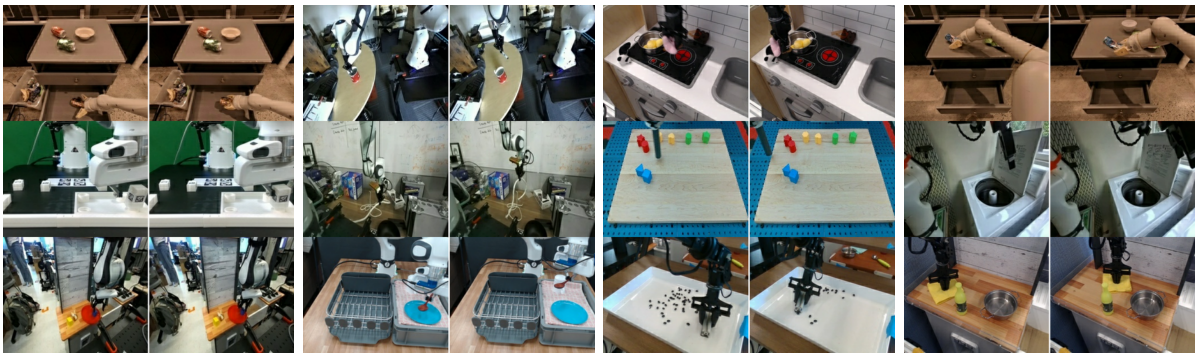


Figure 2: Visualization of image pairs with similar latent actions.

structured information transfer. Our method offers better integration compared to LAPA [67] and GR00T [52], incorporates immediate visual context unlike Moto-GPT [11], and avoids inconsistencies caused by teacher forcing seen in Go-1 [1], achieving robust test-time reasoning.

3 Method

At a high level, our villa-X consists of two main components, as illustrated in Figure 1a:

(i) a LAM (Latent Action Model) module that infers latent actions from a pair of observations;

(ii) a ACT (Actor) module that jointly models latent actions and robot actions given an initial visual observation and a textual task instruction.

We describe each component in detail in the following subsections. Learning in villa-X is carried out in three stages: (1) pretraining the LAM module, (2) pretraining the ACT module, and (3) finetuning ACT on target-embodiment data.

3.1 Latent Action Model

Modeling latent actions enables the use of abundant action-free video demonstration data and also holds the promise of improving generalization between robot and human motions [11, 67]. Building on these prior works, our LAM (depicted in Figure 1b) learns an inverse dynamics model (IDM) to extract latent actions z_t from two adjacent video frames, o_t and o_{t+K}. The interval K is a configurable window size, chosen to balance perceptibility and generalizability: large enough to capture meaningful motion but small enough to maintain transferability.

Previous works [11, 67] train IDM by jointly optimizing a forward dynamics model (FDM) to predict future frames. Given the current frame o_t and a latent action z_t, the model minimizes the discrepancy between the predicted frame $\hat{o}_{t+K}$ and the ground truth o_{t+K}. Although robot data with low-level states and actions are often available alongside action-free video data, these approaches completely discards such information, focusing solely on visual changes. Since our ultimate objective is to use latent actions to facilitate real robots policy learning, bridging vision-language prompts and low-level controls would be beneficial. To this end, we propose augmenting LAM pretraining with proprioceptive supervision. Specifically, since z_t represents a higher-level action abstraction, we introduce an additional proprio FDM module that predicts future robot states and actions K steps ahead given the current robot state q_t and the latent action z_t. This alignment of high-level abstractions and physical robot dynamics enhances the suitability of latent action for robot policy learning.

However, because robot trajectories are often embodiment specific, conditioning solely on q_t and z_t may embed platform-dependent features in the latent action, hindering cross-embodiment transfer. To alleviate this issue, we provide the proprio FDM with an additional embodiment context input (*e.g.*, dataset ID, control frequency), isolating transferable action representations from embodiment particulars.

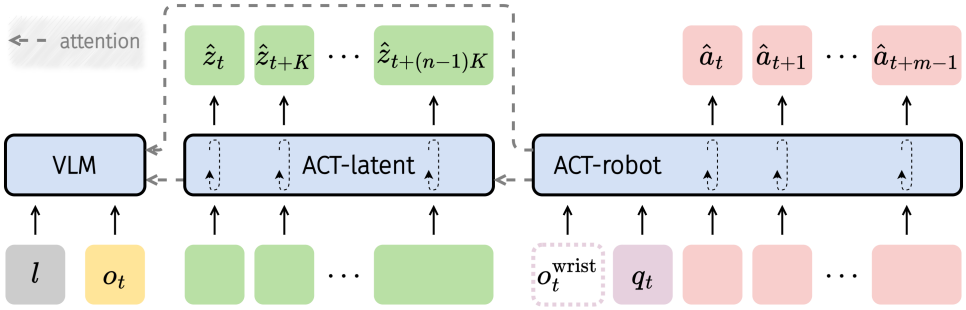


Figure 3: Structure of the ACT module.

Implementation In practice, instead of predicting a single latent action between two frames (as Figure 1b depicts), LAM receives a sequence of T_{LAM} observation frames as input and outputs $T_{\text{LAM}} - 1$ latent actions between consecutive frames. The inverse dynamics model first encodes the frame sequence with a Spatial–Temporal Transformer [65], then applies value quantization to produce $T_{\text{LAM}} - 1$ latent actions $z \in \mathbb{R}^{N_z \times D_d}$. The FDM is implemented using a Vision Transformer (ViT), while the proprio FDM is a two-layer MLP with dual output heads. The embodiment context input is embedded using learnable embeddings and concatenated with the robot state before being fed into the proprio FDM. For further implementation details and configuration specifics, please refer to the appendix.

Training and Inference We train LAM on diverse data sources, including robot trajectories and human egocentric videos. For data lacking low-level state and action labels, we omit the proprio FDM branch and optimize only the FDM reconstruction loss. By leveraging this diverse data, our goal is to learn a generalizable intermediate representation for downstream policy learning. Once trained, only the IDM is required to extract latent actions from a pair of frames at inference, the FDM and proprioceptive FDM serve primarily as diagnostic visualization tools. We use a mixture of datasets containing both human video and robot action data including OpenX [13], Ego4D [22], etc. For more details, please refer to Appendix A.1.

3.2 Actor Module

In the same vein as common VLA models [5, 30, 31, 52, 67], our ACT module builds upon a pre-trained vision-language model (VLM) and is trained to model a generalizable robot control policy that outputs actions given a textual task instruction and an initial observation. With the pre-trained LAM, we are able to incorporate latent actions as an additional intermediate representation in our framework, rather than directly mapping vision and language inputs to low-level robot actions. The latent actions serve as a bridging language that connects task prompts to robot action planning.

Figure 3 provides an illustration of our ACT module, which consists of three components: VLM, ACT-latent, and ACT-robot. Given a textual task prompt l and a visual observation o_t, the VLM first extracts visual and language features. Then, the ACT-latent component, trained with a diffusion objective, models a sequence of n latent actions $z_t, z_{t+K}, \ldots, z_{t+(n-1)K}$, conditioned on the features from the VLM. Subsequently, the ACT-robot component, also implemented as a diffusion model, predicts a sequence of m low-level robot actions, conditioned on the same visual and language features as well as the latent actions. Conditioning across components is achieved using uni-directional attention over intermediate features, as indicated by the dashed line in Figure 3. Additionally, ACT-robot can incorporate wrist camera input when available. Unlike prior work [5], which feeds wrist camera data into the VLM from the start, our design is more computationally efficient. Another consideration is that we view the wrist camera input as more embodiment-specific and less generalizable, which is why we incorporate it into ACT-robot rather than the VLM. The first two components, VLM and ACT-latent, are designed to handle the more generalizable aspects of the policy.

One might ask whether the length of the robot action sequence must match that of the latent action sequence, *i.e.*, whether $m = nK$. In our design, this is not required. We allow different lengths for latent and robot action sequences, as long as they share the same starting time step t. For example, we may model 10 latent actions (corresponding to 20 robot actions if $K = 2$) but choose to predict only 4 robot

actions (effectively relying on the first 2 latent actions). This flexibility enables us to maximize the benefits of modeling latent actions.

We highlight three key elements of our design for incorporating latent actions:

- **Latent actions as a mid-level bridge.** Previous works, including LAPA [67] and GR00T [52], treat latent actions similarly to robot actions and do not explicitly model the hierarchical structure between them. For example, GR00T simply treats latent actions as an additional action type and trains them alongside robot actions within a shared diffusion transformer. In contrast, our model treats latent actions as a distinct mid-level representation that bridges high-level vision and language prompts with low-level robot actions.
- **Explicit transfer from latent to robot actions.** Unlike LAPA [67] and GR00T [52], where the transfer from latent actions to robot actions occurs only implicitly through pre-trained weight initialization, our model enables an explicit connection. The robot action diffusion process is directly conditioned on the latent action diffusion process, allowing more effective and structured information transfer from latent actions to robot actions.
- **Modeling latent actions sequences.** While prior works (LAPA [67] and GO-1 [1]) incorporate latent actions, they only model one-step predictions rather than planning over longer horizons at the latent action level. In contrast, our approach models a sequence of future latent actions, enabling structured planning at both the latent and robot action levels.

Implementation We use a pre-trained PaliGemma model [3] as our VLM. Both `ACT-latent` and `ACT-robot` adopt a transformer architecture following [69], and are trained using flow matching [39, 42]. Wrist camera images are tokenized following the approach used in HPT [63]. Similar to the proprio FDM in `LAM`, we incorporate embodiment context embeddings in `ACT-robot` to account for embodiment-specific information. Inspired by Moto [11] and RDT [43], we apply attention masking during training: with 50% probability, we completely disable attention from latent actions to robot actions; otherwise, we randomly mask out attention from 50% of the latent actions. Further implementation details are provided in the appendix.

Training, Finetuning and Inference During training, we update all components of the model, including both the pre-trained and randomly initialized parts. The model is trained jointly using losses on both latent actions and robot actions, leveraging a mixture of datasets containing both human video and robot action data including OpenX [13], Ego4D [22], etc. Finetuning follows the same procedure as training, with the exception that, for unseen embodiments, we initialize a new context embedding. For robot action inference, we perform denoising steps simultaneously in both `ACT-Latent` and `ACT-Robot`. For more details, please refer to Appendix A.2.

4 Experiments

In this section, we aim to answer the following questions through experiments:

- Does our improved `LAM` learn higher-quality latent actions?
- Can `ACT-latent` successfully plan future motions?
- Can the actor module effectively leverage the pre-trained latent actions?
- How does `villa-X` compare to existing VLA baselines in both simulated benchmarks and real-world robot tasks?

4.1 Does our improved `LAM` learn higher-quality latent actions?

In this subsection, we evaluate whether our improved latent action modeling enhances the quality of the learned latent actions. The key component of our `LAM` is the incorporation of the proprio FDM module. To assess its impact, we compare our model (denoted `w/pp`) to a variant without the proprio FDM module (denoted `wo/pp`).

Probing First, a core expectation for latent actions is that they should carry information useful for predicting low-level robot actions. To test this, we conduct a probing experiment. Specifically, after training the latent action models, we freeze them and train a simple 3-layer MLP to predict the corresponding robot actions for each latent action. Probing is conducted on the LIBERO dataset [40],

Table 1: Evaluation results on SIMPLER for different variants of our villa-X (top group) and alternative approaches for incorporating latent actions (bottom group). "Ours" refers to the w/pp described in the main text.

Method	Google robot					WidowX robot				
	Pick	Move	Drawer	Place	Avg.	Carrot	Eggplant	Spoon	Cube	Avg.
Ours	**81.7**	**55.4**	38.4	4.2	**44.9**	24.2	**71.7**	**48.3**	**19.2**	**40.8**
wo/pp	77.0	52.7	**42.6**	2.8	43.8	22.5	57.5	43.3	5.9	32.3
wo/LAM	42.1	24.6	38.4	0.0	26.3	**25.8**	60.8	36.7	9.2	33.1
LAPA-style	64.7	28.8	38.0	**5.6**	34.2	0.8	0.0	2.5	0.8	1.0
Go-1-style	29.0	38.0	31.3	4.6	25.7	5.8	50.8	1.7	1.0	14.8

which is not used for training latent action models. We train the MLP on the training split of LIBERO and evaluate it using the L1 loss on the validation split.

We are interested in how closely the predicted actions match the ground truth. In LIBERO, the robot action space has 8 dimensions: 3 for position, 4 for rotation, and 1 for the gripper. Rather than averaging the error across dimensions, we focus on the maximum L1 error across all action dimensions, as we want to avoid large deviations in any single aspect of the action. For each model variant (w/pp and wo/pp), we compute the number of validation samples whose maximum L1 error falls below a threshold. By sweeping this threshold, we count how many samples fall within each error bin. A better model should yield more samples with low errors.

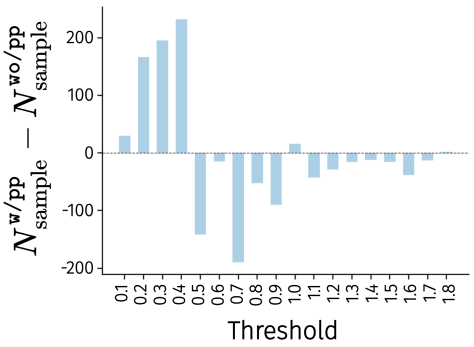


Figure 4: Probing experiment results.

For each error bin, we compute the difference in the number of samples between the w/pp and wo/pp variants and present the results in Figure 4. The w/pp variant produces more samples with smaller errors, while the wo/pp variant has more samples in the high-error bins. This demonstrates the effectiveness of the proprio FDM module in capturing information from the robot actions.

Policy Pre-training Next, we compare how the latent actions generated by different variants of LAM (w/pp and wo/pp) influence policy pre-training. Unlike the main experiments, we pretrain models in this section on a mixture of 10% Fractal [6] data, 10% Bridge V2 [18] data, and 100% Something-Something V2 [20] data, to reduce computation cost while remaining a setting where limited robot data are available for training the VLA model. The resulting policies are evaluated in the SIMPLER environment [32], a simulation benchmark explicitly designed to mitigate the gap between simulated and real-world robotic environments. It comprises two platforms, namely Google robot and WidowX robot, each with four manipulation tasks. We evaluate our method on all the 8 tasks on the visual matching setting. The results are summarized in Table 1. We observe that w/pp clearly outperforms wo/pp, demonstrating the effectiveness of incorporating the proprio FDM module. Additionally, we include a baseline that does not use latent actions (denoted wo/LAM) and is trained solely to predict robot actions. The performance of wo/LAM is significantly worse, indicating that pre-training with latent actions is essential.

LAM visualization Figure 2 visualizes image pairs sharing the same latent action, demonstrating that these pairs correspond to similar underlying robot behaviors. To further demonstrate the transfer ability of our LAM, we extract latent actions from arbitrary video sequences, map them to robot actions using the proprio FDM, and execute the resulting robot action in SIMPLER simulator. The simulated motions closely reproduce the original demonstrations, indicating that latent actions are both aligned with and grounded in the robot's actions. For more examples, please check Appendix C.1.

LAM transfer consistency We provide two visualizations to further evaluate the transfer consistency of the learned latent actions. Specifically, we first use the trained LAM to extract a sequence of latent actions from a source video. These latent actions are then applied to different initial images, allowing a

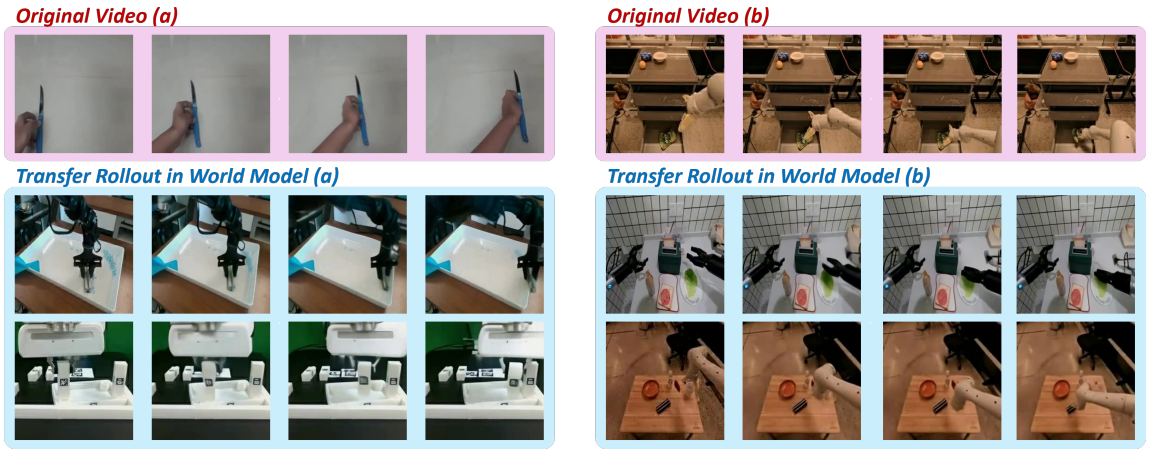


Figure 5: Transfer of human demonstrations to new scenes via latent actions and a world model. (a) moving right. (b) moving downwards. Please visit our website for more demos.

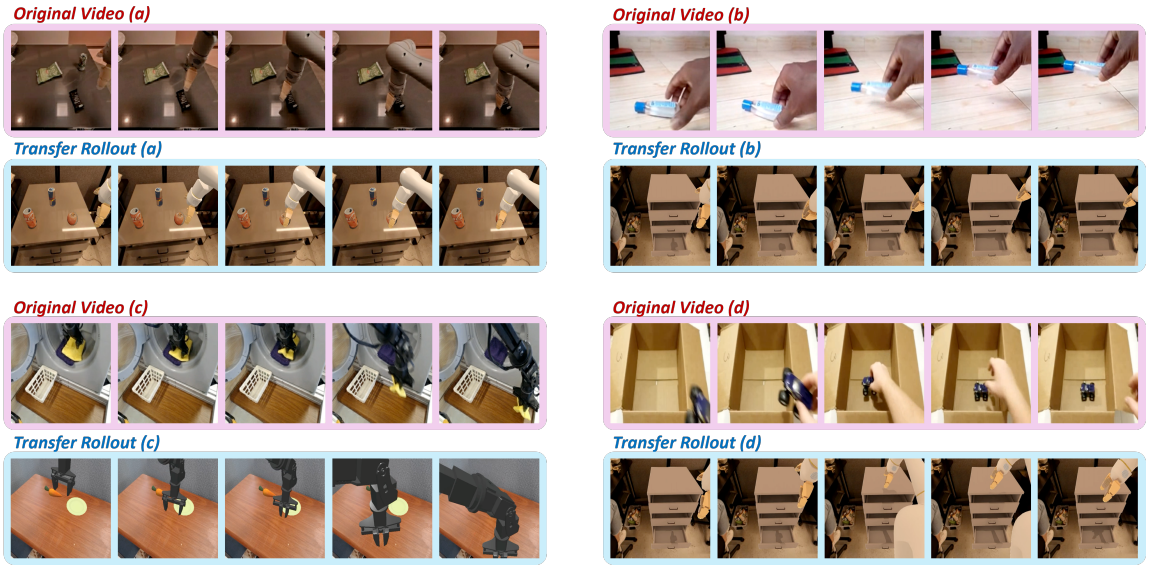


Figure 6: Transfer of video demonstrations into robot action executions via latent actions and a proprio FDM. Please visit our website for more demos.

world model to generate subsequent frames conditioned on these actions. As shown in Figure 5, the transfer rollouts (highlighted in blue) successfully identify the movable parts and demonstrate behaviors consistent with the original video.

We also evaluate the alignment between latent and robot actions. To do this, we use the proprio FDM to decode latent actions into executable robot actions. These actions are then executed in the SIMPLER simulator, with the resulting rollouts visualized in Figure 6. In each example, the top row shows the original video demonstration from which latent actions were extracted, while the bottom row displays the corresponding simulated execution. The simulated motions closely match the original demonstrations, confirming that the latent actions learned by villa-X are well-aligned with and grounded in the robot actions.

4.2 Can ACT-latent successfully plan future motions?

In this experiment, we demostrate the motion planning capabilities of ACT-latent by visualizing its planned actions with a world model. For a given initial image and language instruction, ACT-latent generates a sequence of latent actions, and the world model takes the inital frame and latent actions as inputs to render the latent actions into planned future videos. Figure 7 shows results for both in-

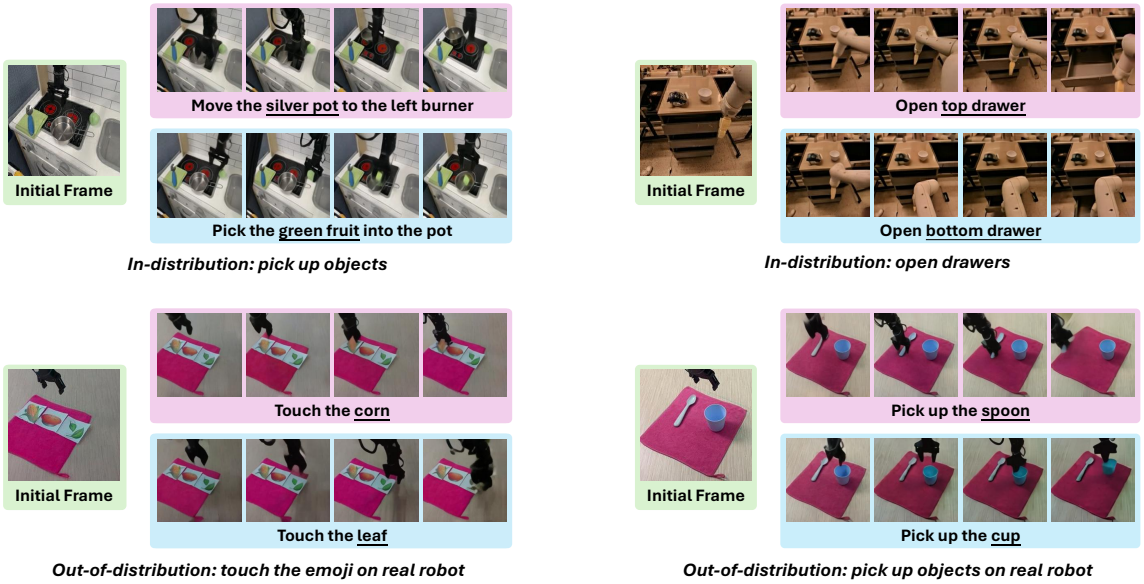


Figure 7: Transfer of video demonstrations into robot action executions via latent actions and a proprio FDM. Please visit our website for more demos.

distribution and out-of-distribution samples, where the in-distribution samples are randomly sampled from the validation set but from the same datasets as used in training, and the out-of-distribution samples are from new real-world scenarios. The results demonstrate that `ACT-latent` successfully follows the language instructions to solve the tasks, while accurately identifying target objects and generating latent actions that successfully follow the instructions. `ACT-latent` also successfully identifies the concepts in the emoji, which rarely appears in robot datasets, suggesting that `villa-X` keeps the general vision-language capabilities in the initial VLM model after pre-training.

4.3 Can the actor module effectively leverage the pre-trained latent actions?

Given high-quality latent actions produced by the pre-trained `LAM`, we investigate whether our design can effectively leverage them to pre-train robot control policies. We compare our approach against two recent methods that also utilize latent actions, albeit in different ways: LAPA [67] and GO-1 [1].

To isolate the effect of how latent actions are incorporated, we implement LAPA-style and GO-1-style models based on our architecture for a fair comparison. For the LAPA-style model, we follow a two-stage pre-training protocol: we first train the VLM to predict latent actions, then replace the latent action prediction head with a robot action prediction head and continue training on data with robot action labels. For the GO-1-style model, we implement a separate latent planner that autoregressively predicts latent actions. The robot action prediction component remains largely unchanged as in our main design.

Following the experiment setup in the previous subsection, we train all models on the same dataset mixture and then evaluate the resulting policies in the SIMPLER environment [32]. The results are shown in Table 1. Compared to other two approaches, our method achieves significantly higher performance, validating the effectiveness of our design for incorporating latent actions into VLA pre-training.

4.4 Evaluating `villa-X` in Simulation

4.4.1 SIMPLER Benchmark

Baselines and Experimental Setup We use the SIMPLER benchmark as described above. In this section, we compare against two categories of prior work:

- Vision-Language-Action (VLA) models: RT-1-X [13], Octo-base [54], OpenVLA [30], RoboVLMs [34], which learn policies solely from mixed robot datasets.
- Latent-Action based methods: GR00T [53], MoTo [11] and LAPA [67], which additionally exploit unlabelled videos by inferring latent actions.

Table 2: Comparison on SIMPLER of villa-X and existing methods. Methods marked with $*$ are evaluated directly after pretraining, whereas other methods are evaluated after post-training on corresponding dataset.

Method	Google Robot					WidowX Robot				
	Pick	Move	Drawer	Place	Avg.	Carrot	Eggplant	Spoon	Cube	Avg..
RT-1-X $*$	56.7	31.7	**59.7**	21.3	42.4	4.2	0.0	0.0	0.0	1.1
Octo-base $*$	17.0	4.2	22.7	0.0	11.0	8.3	43.1	12.5	0.0	16.0
OpenVLA $*$	16.3	46.2	35.6	0.0	24.5	0.0	4.1	0.0	0.0	1.0
RoboVLMs $*$	72.7	66.3	26.8	**36.1**	50.5	25.0	0.0	20.8	8.3	13.5
RoboVLMs	77.3	61.7	43.5	24.1	51.7	20.8	**79.2**	45.8	4.2	37.5
GR00T	0.7	1.9	2.9	0.0	1.4	0.0	13.9	1.4	0.0	3.8
MoTo	74.0	60.4	43.1	N/A	N/A	N/A	N/A	N/A	N/A	N/A
LAPA	N/A	N/A	N/A	N/A	N/A	45.8	58.3	70.8	54.2	57.3
Ours w/o latent	56.3	25.8	27.3	13.9	30.8	31.3	74.6	61.7	28.3	49.0
Ours	**98.7**	**75.0**	**59.3**	5.6	**59.6**	**46.3**	64.6	**77.9**	**61.3**	**62.5**

Except where noted ($*$), all models follow a two-stage pretraining–finetuning protocol, including a general pretraining phase on large-scale mixed data, followed by finetuning on a dataset of specific embodiment. We also include an ablation (villa-X w/o latent) that removes our latent-action expert while keeping all other components unchanged. Reported baseline scores are taken directly from the original publications, while missing entries are marked as N/A.

Experimental Results Table 2 summarizes the success rates on both platforms. Our full model achieves the highest score on average success rate on the Google robot (59.6%) and the WidowX robot (62.5%). This improvement over VLA methods, which cannot exploit unlabelled video, demonstrates the benefit of our incorporating human videos into policy learning. Moreover, our approach outperforms other latent-action methods, indicating that our specific mechanism for leveraging video data is more effective. Finally, the gap between our full model and the "villa-X w/o latent" ablation confirms that the latent-action expert is essential for achieving these gains.

4.4.2 LIBERO Benchmark

The LIBERO benchmark [40] evaluates knowledge transfer in multitask and lifelong robot learning problems for robotic manipulation, consisting of four task suites: **LIBERO-Spatial** evaluates the model's performance under novel layouts with the same task and object types, **LIBERO-Goal** evaluates the model's performance under novel tasks with the same object types and layouts, **LIBERO-Object** evaluates the model's performance under novel object types with the same tasks and layouts, **LIBERO-Long** evaluates the model's performance under diverse set of objects, layouts and backgrounds. Each task suite contains 10 tasks with 50 human demonstrations per task for fine-tuning.

Baselines and Experimental Setup We compare with the following existing models: Diffusion Policy [12] trained from scratch, Octo [54], and OpenVLA [30]. All models follow a two-stage pretraining-finetuning protocol. We finetune villa-X and villa-X w/o latent on the demonstration data of the each task suite separately, and test on the LIBERO simulator for 10 tasks and 20 trials per task on each task suite.

Experimental Results Table 3 summarizes the success rates on each task suite of LIBERO. Our model achieves better performance than existing methods in all the four task suites. Also, our model with latent action achieves higher performance on 3 of the 4 task suites and average performance, confirming that the proposed latent action expert improves the manipulation performance.

Table 3: Evaluation on 4 LIBERO task suites of villa-X and existing methods.

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy [12]	78.3	92.5	68.3	50.5	72.4
Octo-base [54]	78.9	85.7	84.6	51.1	75.1
OpenVLA [30]	84.7	88.4	79.2	53.7	76.5
Ours w/o latent	86.0	86.5	85.0	70.0	81.9
Ours	**97.5**	**97.0**	**91.5**	**74.5**	**90.1**

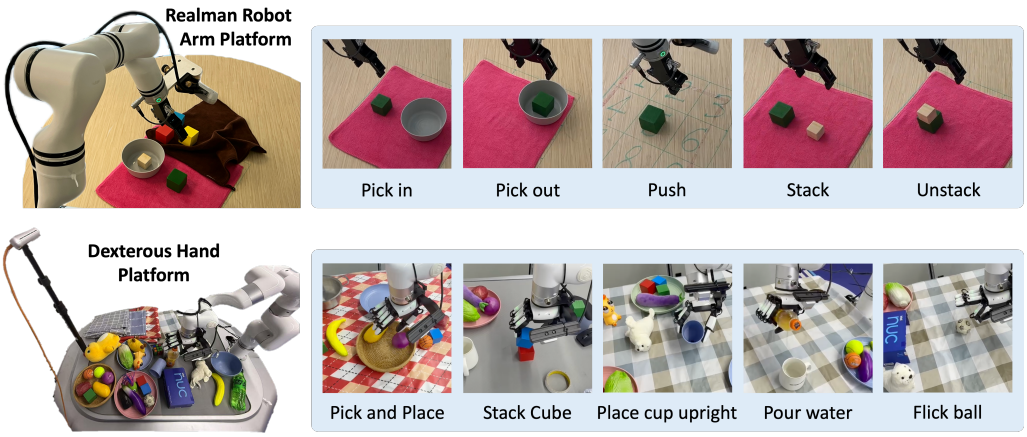


Figure 8: Real-world robot evaluation platforms: **(top)** Realman robot arm platform with a gripper and **(bottom)** Xarm robot arm with Xhand dexterous hand. Platform setups are shown on the left, with corresponding evaluation tasks on the right. Please visit our website for more videos.

4.5 Evaluating villa-X on Real-world Robots

To further assess generalization, we evaluate our approach on two real-world robotic platforms: a Realman robot arm with a gripper, and an Xarm manipulator equipped with a 12-DOF Xhand dexterous hand. See Figure 8 for details.

Realman robot arm with gripper On the robot-arm manipulation platform, we use a 6-DoF Realman RM 75 robot arm and a 1-DoF Inspire gripper. We fine-tune and evaluate our policy on "Pick-in" (pick the block into a bowl), "Pick-out" (pick the block out of a bowl), "Stack" (stack the block onto another block), "Unstack" (unstack the block from another block), and "Push" (push the block to a given location) tasks. For fine-tuning, we collect a dataset of 375 trajectories through teleoperation (with 75 trajectories for each task), where the object layout and table setup are fixed and only the object locations are dynamic.

We conduct two sets of evaluation: In task evaluation, we remain the table setup the same as data collection; in generalization evaluation, we change the color of the block and table cover. We evaluate the policy with 10 trials per task with each trial conducted with different object location, where all the experiment settings are the same for the evaluation of different policies (including the object location and lightening condition). The results are shown in Table 5, demonstrating that villa-X outperforms existing baselines in both task and generalization evaluation sets.

Xarm robot arm with Xhand dexterous hand On the dexterous-hand platform, we use the Xhand, a 12-DoF dexterous hand with five flexible fingers, mounted on a 7-DoF Xarm robot arm. Fine-tuning is performed on the Xhand Dataset [26], which comprises 4,000 trajectories spanning 13 task categories. Since no dexterous-hand data were used during pretraining, this evaluation can test embodiment transfer ability. We select five representative tasks—pick-and-place, cube stacking, cup upright placement, water pouring and ball flicking. The results are summarized in Table 4 for (i) seen tasks, where objects are randomly replaced or additional distractors are added, and (ii) unseen tasks, which use unseen objects or backgrounds. The performances are evaluated under 50 runs for pick and place, 20 runs for stack cube and 10 runs for others. Table 4 demonstrates that our method outperforms existing baselines.

Table 4: Evaluation on Xarm robot arm of villa-X and existing methods.

Method	Pick & Place		Stack Cube		Place Cup Upright		Pour Water		Flick Ball	
	seen	unseen	seen	unseen	seen	unseen	seen	unseen	seen	unseen
GR-1	56	40	15	5	0	0	0	0	40	10
GR00T	44	28	20	0	20	0	0	0	30	0
Ours w/o latent	72	60	70	40	40	30	40	10	50	30
Ours	**84**	**68**	**75**	**50**	**60**	**30**	**60**	**30**	**50**	**40**

Table 5: Evaluation on Realman robot arm of villa-X and existing methods.

Method	Pick in	Pick out	Push	Stack	Unstack	Change block color	Change table cover
GR00T	30	70	10	10	60	50	30
Ours w/o latent	**40**	80	30	**60**	70	40	30
Ours	30	**100**	**50**	50	**100**	**60**	**60**

5 Conclusion, Limitations, and Future Works

In this paper, we presented villa-X, a novel Visual-Language-Latent-Action (ViLLA) framework that improves both the learning of latent actions and their incorporation into VLA pre-training. Our experiments demonstrate that our enhanced Latent Action Model learns higher-quality latent actions, and our improved policy model more effectively leverages these learned actions. Overall, our method exhibits superior performance in both simulated environments and real-world robotic tasks.

One limitation is that the proposed latent expert, although effective at future planning through both visual and proprioceptive state planning, is not fully explored in this work. For example, future research could learn a critic with prior knowledge from foundation vision-language models, allowing multiple samples from the latent expert and rejecting planned trajectories that do not follow the language instruction. We leave this aspect as future work to further improve the capability of the ViLLA framework.

References

[1] AgiBot-World-Contributors, Bu, Q., Cai, J., Chen, L., Cui, X., Ding, Y., Feng, S., Gao, S., He, X., Huang, X., Jiang, S., Jiang, Y., Jing, C., Li, H., Li, J., Liu, C., Liu, Y., Lu, Y., Luo, J., Luo, P., Mu, Y., Niu, Y., Pan, Y., Pang, J., Qiao, Y., Ren, G., Ruan, C., Shan, J., Shen, Y., Shi, C., Shi, M., Shi, M., Sima, C., Song, J., Wang, H., Wang, W., Wei, D., Xie, C., Xu, G., Yan, J., Yang, C., Yang, L., Yang, S., Yao, M., Zeng, J., Zhang, C., Zhang, Q., Zhao, B., Zhao, C., Zhao, J., and Zhu, J. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv: 2503.06669*, 2025.

[2] Belkhale, S., Cui, Y., and Sadigh, D. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

[3] Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bošnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Henaff, O., Xiong, X., Soricut, R., Harmsen, J., and Zhai, X. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv: 2407.07726*, 2024.

[4] Beyer, L., Steiner, A., Pinto, A. S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., Unterthiner, T., Keysers, D., Koppula, S., Liu, F., Grycner, A., Gritsenko, A., Houlsby, N., Kumar, M., Rong, K., Eisenschlos, J., Kabra, R., Bauer, M., Bošnjak, M., Chen, X., Minderer, M., Voigtlaender, P., Bica, I., Balazevic, I., Puigcerver, J., Papalampidi, P., Henaff, O., Xiong, X., Soricut, R., Harmsen, J., and Zhai, X. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[5] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K.,

Shi, L. X., Tanner, J., Vuong, Q., Walling, A., Wang, H., and Zhilinsky, U. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv: 2410.24164*, 2024.

[6] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N. J., Julian, R. C., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.-H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P. R., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-1: Robotics transformer for real-world control at scale. *Robotics: Science and Systems*, 2022. doi: 10.48550/arXiv.2212.06817.

[7] Bruce, J., Dennis, M. D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

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

[9] Chen, L. Y., Adebola, S., and Goldberg, K. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

[10] Chen, X., Guo, J., He, T., Zhang, C., Zhang, P., Yang, D. C., Zhao, L., and Bian, J. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

[11] Chen, Y., Ge, Y., Li, Y., Ge, Y., Ding, M., Shan, Y., and Liu, X. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv: 2412.04445*, 2024.

[12] Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

[13] Collaboration, O. X.-E., O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., Tung, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Gupta, A., Wang, A., Kolobov, A., Singh, A., Garg, A., Kembhavi, A., Xie, A., Brohan, A., Raffin, A., Sharma, A., Yavary, A., Jain, A., Balakrishna, A., Wahid, A., Burgess-Limerick, B., Kim, B., Schölkopf, B., Wulfe, B., Ichter, B., Lu, C., Xu, C., Le, C., Finn, C., Wang, C., Xu, C., Chi, C., Huang, C., Chan, C., Agia, C., Pan, C., Fu, C., Devin, C., Xu, D., Morton, D., Driess, D., Chen, D., Pathak, D., Shah, D., Büchler, D., Jayaraman, D., Kalashnikov, D., Sadigh, D., Johns, E., Foster, E., Liu, F., Ceola, F., Xia, F., Zhao, F., Frujeri, F. V., Stulp, F., Zhou, G., Sukhatme, G. S., Salhotra, G., Yan, G., Feng, G., Schiavi, G., Berseth, G., Kahn, G., Yang, G., Wang, G., Su, H., Fang, H.-S., Shi, H., Bao, H., Amor, H. B., Christensen, H. I., Furuta, H., Walke, H., Fang, H., Ha, H., Mordatch, I., Radosavovic, I., Leal, I., Liang, J., Abou-Chakra, J., Kim, J., Drake, J., Peters, J., Schneider, J., Hsu, J., Bohg, J., Bingham, J., Wu, J., Gao, J., Hu, J., Wu, J., Wu, J., Sun, J., Luo, J., Gu, J., Tan, J., Oh, J., Wu, J., Lu, J., Yang, J., Malik, J., Silvério, J., Hejna, J., Booher, J., Tompson, J., Yang, J., Salvador, J., Lim, J. J., Han, J., Wang, K., Rao, K., Pertsch, K., Hausman, K., Go, K., Gopalakrishnan, K., Goldberg, K., Byrne, K., Oslund, K., Kawaharazuka, K., Black, K., Lin, K., Zhang, K., Ehsani, K., Lekkala, K., Ellis, K., Rana, K., Srinivasan, K., Fang, K., Singh, K. P., Zeng, K.-H., Hatch, K., Hsu, K., Itti, L., Chen, L. Y., Pinto, L., Fei-Fei, L., Tan, L., Fan, L. J., Ott, L., Lee, L., Weihs, L., Chen, M., Lepert, M., Memmel, M., Tomizuka, M., Itkina, M., Castro, M. G., Spero, M., Du, M., Ahn, M., Yip, M. C., Zhang, M., Ding, M., Heo, M., Srirama, M. K., Sharma, M., Kim, M. J., Kanazawa, N., Hansen, N., Heess, N., Joshi, N. J., Suenderhauf, N., Liu, N., Palo, N. D., Shafiullah, N. M. M., Mees, O., Kroemer, O., Bastani, O., Sanketi, P. R., Miller, P. T., Yin, P., Wohlhart, P., Xu, P., Fagan, P. D., Mitrano, P., Sermanet, P., Abbeel, P., Sundaresan, P., Chen, Q., Vuong, Q., Rafailov, R., Tian, R., Doshi, R., Mart'in-Mart'in, R., Baijal, R., Scalise, R., Hendrix, R., Lin, R., Qian, R., Zhang, R., Mendonca, R., Shah, R., Hoque, R., Julian, R., Bustamante, S., Kirmani, S., Levine, S., Lin, S., Moore, S., Bahl, S., Dass, S., Sonawani, S., Song, S., Xu, S., Haldar, S., Karamcheti, S., Adebola, S., Guist, S., Nasiriany, S., Schaal, S., Welker, S., Tian, S., Ramamoorthy, S., Dasari, S., Belkhale, S., Park, S., Nair, S., Mirchandani, S., Osa, T., Gupta, T., Harada, T., Matsushima, T., Xiao, T., Kollar, T., Yu, T., Ding, T., Davchev, T., Zhao, T. Z., Armstrong, T., Darrell, T., Chung, T., Jain, V., Vanhoucke, V., Zhan, W., Zhou, W., Burgard, W., Chen, X., Chen, X., Wang, X., Zhu, X., Geng, X., Liu, X., Liangwei, X., Li, X., Pang, Y., Lu, Y., Ma, Y. J., Kim,

Y., Chebotar, Y., Zhou, Y., Zhu, Y., Wu, Y., Xu, Y., Wang, Y., Bisk, Y., Dou, Y., Cho, Y., Lee, Y., Cui, Y., Cao, Y., Wu, Y.-H., Tang, Y., Zhu, Y., Zhang, Y., Jiang, Y., Li, Y., Li, Y., Iwasawa, Y., Matsuo, Y., Ma, Z., Xu, Z., Cui, Z. J., Zhang, Z., Fu, Z., and Lin, Z. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[14] Cui, Z. J., Wang, Y., Shafiullah, N. M. M., and Pinto, L. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[15] Cui, Z. J., Pan, H., Iyer, A., Haldar, S., and Pinto, L. Dynamo: In-domain dynamics pretraining for visuo-motor control. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3b8db54b629e00537b59cbc6612026d7-Abstract-Conference.html.

[16] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020.

[17] Dass, S., Yapeter, J., Zhang, J., Zhang, J., Pertsch, K., Nikolaidis, S., and Lim, J. J. CLVR jaco play dataset, 2023. URL https://github.com/clvrai/clvr_jaco_play_dataset.

[18] Ebert, F., Yang, Y., Schmeckpeper, K., Bucher, B., Georgakis, G., Daniilidis, K., Finn, C., and Levine, S. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[19] Fang, H.-S., Fang, H., Tang, Z., Liu, J., Wang, J., Zhu, H., and Lu, C. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[20] Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] Goyal, R., Kahou, S. E., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense, 2017. URL https://arxiv.org/abs/1706.04261.

[22] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Fuegen, C., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P. R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G. M., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[23] Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

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

[25] Heo, M., Lee, Y., Lee, D., and Lim, J. J. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[26] Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.-J., Zhang, J., Sreenath, K., Lu, C., and Chen, J. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[27] Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

[28] Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, pp. 651–673, 2018.

[29] Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., Fagan, P. D., Hejna, J., Itkina, M., Lepert, M., Ma, Y. J., Miller, P. T., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A., Black, K., Chi, C., Hatch, K. B., Lin, S., Lu, J., Mercat, J., Rehman, A., Sanketi, P. R., Sharma, A., Simpson, C., Vuong, Q., Walke, H. R., Wulfe, B., Xiao, T., Yang, J. H., Yavary, A., Zhao, T. Z., Agia, C., Baijal, R., Castro, M. G., Chen, D., Chen, Q., Chung, T., Drake, J., Foster, E. P., Gao, J., Herrera, D. A., Heo, M., Hsu, K., Hu, J., Jackson, D., Le, C., Li, Y., Lin, K., Lin, R., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T., O'Neill, A., Scalise, R., Seale, D., Son, V., Tian, S., Tran, E., Wang, A. E., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J. J., Malik, J., Martín-Martín, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M. C., Zhu, Y., Kollar, T., Levine, S., and Finn, C. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[30] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[31] Li, Q., Liang, Y., Wang, Z., Luo, L., Chen, X., Liao, M., Wei, F., Deng, Y., Xu, S., Zhang, Y., et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[32] Li, X., Hsu, K., Gu, J., Mees, O., Pertsch, K., Walke, H. R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., and Xiao, T. Evaluating real-world robot manipulation policies in simulation. In Agrawal, P., Kroemer, O., and Burgard, W. (eds.), *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728. PMLR, 2024. URL https://proceedings.mlr.press/v270/li25c.html.

[33] Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H. R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., and Xiao, T. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[34] Li, X., Li, P., Liu, M., Wang, D., Liu, J., Kang, B., Ma, X., Kong, T., Zhang, H., and Liu, H. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

[35] Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., Li, H., and Kong, T. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=lFYj0oibGR.

[36] Li, Y., Liu, M., and Rehg, J. M. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 619–635, 2018.

[37] Li, Y., Cao, Z., Liang, A., Liang, B., Chen, L., Zhao, H., and Feng, C. Egocentric prediction of action target in 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[38] Liang, A., Czempin, P., Hong, M., Zhou, Y., Biyik, E., and Tu, S. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

[39] Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[40] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[41] Liu, H., Nasiriany, S., Zhang, L., Bao, Z., and Zhu, Y. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.

[42] Liu, Q. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

[43] Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., and Zhu, J. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv: 2410.07864*, 2024.

[44] Liu, Y., Liu, Y., Jiang, C., Lyu, K., Wan, W., Shen, H., Liang, B., Fu, Z., Wang, H., and Yi, L. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21013–21022, June 2022.

[45] Luo, J., Xu, C., Liu, F., Tan, L., Lin, Z., Wu, J., Abbeel, P., and Levine, S. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

[46] Lynch, C., Wahid, A., Tompson, J., Ding, T., Betker, J., Baruch, R., Armstrong, T., and Florence, P. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[47] Mees, O., Borja-Diaz, J., and Burgard, W. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[48] Mendonca, R., Bahl, S., and Pathak, D. Structured world models from human videos. *CoRL*, 2023.

[49] Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., and Luo, P. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36:25081–25094, 2023.

[50] Nasiriany, S., Gao, T., Mandlekar, A., and Zhu, Y. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

[51] Nikulin, A., Zisman, I., Tarasov, D., Lyubaykin, N., Polubarov, A., Kiselev, I., and Kurenkov, V. Latent action learning requires supervision in the presence of distractors, 2025. URL https://arxiv.org/abs/2502.00379.

[52] NVIDIA, :, Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L. J., Fang, Y., Fox, D., Hu, F., Huang, S., Jang, J., Jiang, Z., Kautz, J., Kundalia, K., Lao, L., Li, Z., Lin, Z., Lin, K., Liu, G., Llontop, E., Magne, L., Mandlekar, A., Narayan, A., Nasiriany, S., Reed, S., Tan, Y. L., Wang, G., Wang, Z., Wang, J., Wang, Q., Xiang, J., Xie, Y., Xu, Y., Xu, Z., Ye, S., Yu, Z., Zhang, A., Zhang, H., Zhao, Y., Zheng, R., and Zhu, Y. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv: 2503.14734*, 2025.

[53] NVIDIA, :, Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L. J., Fang, Y., Fox, D., Hu, F., Huang, S., Jang, J., Jiang, Z., Kautz, J., Kundalia, K., Lao, L., Li, Z., Lin, Z., Lin, K., Liu, G., Llontop, E., Magne, L., Mandlekar, A., Narayan, A., Nasiriany, S., Reed, S., Tan, Y. L., Wang, G., Wang, Z., Wang, J., Wang, Q., Xiang, J., Xie, Y., Xu, Y., Xu, Z., Ye, S., Yu, Z., Zhang, A., Zhang, H., Zhao, Y., Zheng, R., and Zhu, Y. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[54] Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Xu, C., Luo, J., Kreiman, T., Tan, Y., Chen, L. Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., and Levine, S. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[55] Pei, B., Huang, Y., Xu, J., Chen, G., He, Y., Yang, L., Wang, Y., Xie, W., Qiao, Y., Wu, F., and Wang, L. Modeling fine-grained hand-object dynamics for egocentric video representation learning, 2025. URL https://arxiv.org/abs/2503.00986.

[56] Quere, G., Hagengruber, A., Iskandar, M., Bustamante, S., Leidner, D., Stulp, F., and Vogel, J. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7, Paris, France, 2020.

[57] Rosete-Beas, E., Mees, O., Kalweit, G., Boedecker, J., and Burgard, W. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[58] Schmidt, D. and Jiang, M. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023.

[59] Shafiullah, N. M. M., Rai, A., Etukuru, H., Liu, Y., Misra, I., Chintala, S., and Pinto, L. On bringing robots home, 2023.

[60] Walke, H., Black, K., Lee, A., Kim, M. J., Du, M., Zheng, C., Zhao, T., Hansen-Estruch, P., Vuong, Q., He, A., Myers, V., Fang, K., Finn, C., and Levine, S. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[61] Wang, J., Zhang, Q., Chao, Y.-W., Wen, B., Guo, X., and Xiang, Y. Ho-cap: A capture system and dataset for 3d reconstruction and pose tracking of hand-object interaction, 2024. URL https://arxiv.org/abs/2406.06843.

[62] Wang, L., Chen, X., Zhao, J., and He, K. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 124420–124450. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e0f393e7980a24fd12fa6f15adfa25fb-Paper-Conference.pdf.

[63] Wang, L., Chen, X., Zhao, J., and He, K. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Neural Information Processing Systems*, 2024. doi: 10.48550/arXiv.2409.20537.

[64] Wang, X., Kwon, T., Rad, M., Pan, B., Chakraborty, I., Andrist, S., Bohus, D., Feniello, A., Tekin, B., Frujeri, F. V., Joshi, N., and Pollefeys, M. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20270–20281, October 2023.

[65] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv: 2001.02908*, 2020.

[66] Yang, J., Shi, Y., Zhu, H., Liu, M., Ma, K., Wang, Y., Wu, G., He, T., and Wang, L. Como: Learning continuous latent motion from internet videos for scalable robot learning, 2025. URL https://arxiv.org/abs/2505.17006.

[67] Ye, S., Jang, J., Jeon, B., Joo, S., Yang, J., Peng, B., Mandlekar, A., Tan, R., Chao, Y.-W., Lin, B. Y., Liden, L., Lee, K., Gao, J., Zettlemoyer, L., Fox, D., and Seo, M. Latent action pretraining from videos. *arXiv preprint arXiv: 2410.11758*, 2024.

[68] Zhao, Q., Lu, Y., Kim, M. J., Fu, Z., Zhang, Z., Wu, Y., Li, Z., Ma, Q., Han, S., Finn, C., Handa, A., Liu, M.-Y., Xiang, D., Wetzstein, G., and Lin, T.-Y. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv: 2503.22020*, 2025.

[69] Zhou, C., Yu, L., Babu, A., Tirumala, K., Yasunaga, M., Shamis, L., Kahn, J., Ma, X., Zettlemoyer, L. S., and Levy, O. Transfusion: Predict the next token and diffuse images with one multi-modal model. *International Conference on Learning Representations*, 2025.

A Model Design

A.1 Latent Action Model

There are four modules in our latent action model. A ST-Transformer based inverse dynamic model (IDM), a vector quantization module, an image reconstruction forward dynamic model (FDM) and a robot proprio FDM. The IDM takes an $8 \times 224 \times 224$ video clip as input, following a Patch Embedding with a patch size of 14 and 12 ST-blocks, where each block has a hidden dimension of 768 and 32 attention heads. The codebook size of vector quantization is set to 32. Image reconstruction FDM is a 12-layer ViT-base network, and robot proprio FDM is a 2-layer MLP with a hidden size of 768. Our model is trained with a learning rate of 1.5e-4, a batch size of 512, and a 2000-step linear warmup. We use the same loss weight for the image FDM and proprio FDM. The pretraining takes 4 days on 128 NVIDIA A100 GPUs.

A.2 Actor Module

Our VLA model comprises three components. First, the vision–language encoder is based on PaliGemma [4], a 3B-parameter VLM pretrained with 224×224 images and 128-token text inputs. Second and third, the latent-action expert and the robot-action expert are each implemented as 18-layer Transformer networks, mirroring PaliGemma's design, with a hidden dimension of 1,024 and 8 attention heads. For the latent action sequence, we select a sequence length of $N = 6$, and for the robot actions, we select a sequence length of $M = 4$.

We extend our policy head with a variant of HPT [62], assigning each embodiment its own pair of state- and action-projection layers while sharing all other parameters. Visual features from the wrist camera are extracted by a pretrained ResNet-18 [24] and fused into the main model via a shared cross-attention head that maps the ResNet features into 16 tokens. During training, wrist-view inputs are randomly masked 50% of the time. We also observed that the latent-action representation can be overly exploited by the robot-action expert, so we regularize this with two complementary dropout schemes. First, we add a 50% attention-weight dropout on the latent-action stream. For the remaining tokens, we randomly mask 50% latent action tokens. This combined masking strategy encourages the model to learn robust, generalizable policy that will balance the predicted latent actions as well as the input image and instruction. Each expert contains approximately 300 M parameters and is trained from scratch. We train all components jointly using a learning rate of $5e-5$ with a 200-step linear warmup. We clip gradients to a maximum norm of 1.0 to ensure stable optimization. The pretraining takes 4 days on 64 NVIDIA A100 GPUs.

B Dataset

B.1 Data Mixture

We curated a data mixture by combining both robot data and action-free human videos for our pretraining phase. For robot data, we draw primarily from OpenX [13] mixture and AgiBot [1]. For OpenX dataset, our base data mixture is created primarily based on [30, 54]. In total, we use 223.5M trajectories with 1.6M frames of robot data. For human videos, we use a mixture of Ego4D [22], EgoPAT3D [37], EGTEA Gaze+ [36], EPIC-KITCHENS [16], HO-Cap [61], HOI4D [44], HoloAssist [64], RH20T [19], Something Something V2 [21]. Altogether, this yields 3.6M clips of human videos. During LAM pretraining, we exclusively utilize the primary third-person camera view. For policy pretraining, we optionally incorporate the wrist-mounted view (when available), applying a 50% dropout. A full breakdown of our data mixture is listed in Table 6.

B.2 Data Preprocessing

For data cleaning, we adopt EgoHOD [55], a curated subset of Ego4D [22], and further filter the videos based on visual quality to ensure high-quality inputs for training. For both robot data and human videos, we apply random adjustments to brightness, contrast, saturation, and hue as data augmentation. In the case of robot data, we represent both proprioceptive states and actions using euler angles.

Dataset	Mix Ratio (%)
RT-1 Robot Action [6]	9.70
AgiBot World Beta [1]	20.0
Kuka [28]	1.97
Bridge [18, 60]	5.47
Taco Play [47, 57]	0.76
Jaco Play [17]	0.12
Berkely Autolab UR5 [9]	0.31
Language Table [46]	0.11
Stanford Hydra Dataset [2]	1.61
NYU Franka Play Dataset [14]	0.22
Furniture Bench Dataset [25]	0.63
Austin Sailor Dataset [50]	0.57
Austin Sirius Dataset [41]	0.45
BC-Z [27]	3.47
DLR EDAN Shared Control [56]	0.01
CMU Stretch [48]	0.04
FMB Dataset [45]	0.73
DobbE [59]	0.37
DROID [29]	3.46
Ego4D [23, 55]	21.46
EgoPAT3D [37]	0.94
EGTEA Gaze+ [36]	0.89
EPIC-KITCHENS [16]	6.95
HO-Cap [61]	0.63
HOI4D [44]	1.99
HoloAssist [64]	4.77
RH20T [19]	5.56
Something-Something V2 [20]	6.82

Table 6: Our training data mixture used during the pretraining phase.

C LAM visualization results

C.1 Image Pairs with Similar Latent Actions

Figure 9 visualizes additional image pairs sharing the same latent action, demonstrating that these pairs correspond to similar underlying robot behaviors.

The results demonstrate that similar latent actions represent the similar robot behaviors and low-level actions, in regardless of which embodiment (including human and different robots) is executing such action. This results support that villa-X learns cross-embodiment prior knowledge for manipulations with latent actions.

D Simulation Evaluation Details

D.1 SIMPLER Benchmark

We evaluate on all eight SIMPLER [33] tasks in the visual matching setting, which include two robot platforms: Google Robot and WidowX.

For Google Robot, the tasks are: (1) pick coke can (including horizontal, vertical and standing can configurations); (2) move an object near a target object; (3) open / close top, middle or bottom drawer; and (4) place apple in a closed drawer, which includes two subtasks: first open top drawer, and then place the apple into the top drawer. On the widowX setup, the tasks consist of: (1) put a carrot on the plate; (2) put an eggplant on the basket; (3) put a spoon on the towel; (4) stack a green cube on a yellow one.

Figure 9: Visualization of image pairs with similar latent actions.

We follow the standard evaluation protocol to test by randomizing both configurations of the environments. For the Google Robot tasks, we execute 300 trials of "Pick Coke Can", 240 of "Move Near", 216 of "Open/Close Drawer", and 108 of "Place Apple in Closed Drawer". For each WidowX task, we use 24 unique configurations. To ensure statistical significance, we test each configuration 10 times, yielding 240 rollouts per task. Reported results (Table 2) are the average success rates across these trials. Please refer to SIMPLER [33] for more details.

For a fair comparison, we adopt the published performance metrics for RT-1-X [13], Octo-base [54], OpenVLA [30], RoboVLMs [34], MoTo [11], and LAPA [67] directly from their respective papers. In the case of GR00T [53], we use the official pretrained checkpoint and performe fine-tuning on the RT-1/Bridge dataset following the authors' published guidelines accordingly.

D.2 LIBERO

The LIBERO benchmark [40] evaluates knowledge transfer in multitask and lifelong robot learning problems for robotic manipulation, consisting of four task suites: **LIBERO-Spatial**, **LIBERO-Goal**, **LIBERO-Object**, **LIBERO-Long**. For fine-tuning of our models, we reinitialize the linear state encoder, action encoder, and action decoder, and tune the full parameters (except for the vision encoder). We fine-tune all the models for 50k gradient steps on the fine-tuning dataset on each task suite of LIBERO. We follow the standard evaluation protocol to test by randomizing the initial states of the environments. Reported results (Table 3) are the average success rates across on a total of 200 trials and a total of 20 trials per task on each task suite. Please refer to Section 4.2 and Appendix C of LIBERO [40] for more details.

E Real-world Robot Platforms Evaluation Details

E.1 Realman robot arm

The Realman robot arm setup is shown in Figure 8 (upper). We mount the gripper for Inspire Robot to the Realman RM75 robot arm. We use two camera views, including a primary view camera with the same view point as the images (used to demonstrate different tasks) shown in Figure 8 (upper) and a wrist camera. For fine-tuning of our models, we reinitialize the linear state encoder, action encoder, and action decoder, and tune the full parameters (except for the vision encoder). We fine-tune all the models for 60k gradient steps.

We collect data on the following five tasks with their task instructions:

- Put-in: "Pick the green block from the table into the blue bowl"
- Put-out: "Pick the green block from the blue bowl onto the table"

- Push: "Push the green block to position X" where "X" indicates the nine positions written on the table.
- Stack: "Stack the wooden block onto the green block"
- Unstack: "Unstack the wooden block from the green block"

We collect 375 trajectories (75 trajectories for each task) for fine-tuning. The trajectories are collected at 10Hz. We post-process these trajectories to remove static frames with zero action, resulting in 120 steps on average in one trajectory.

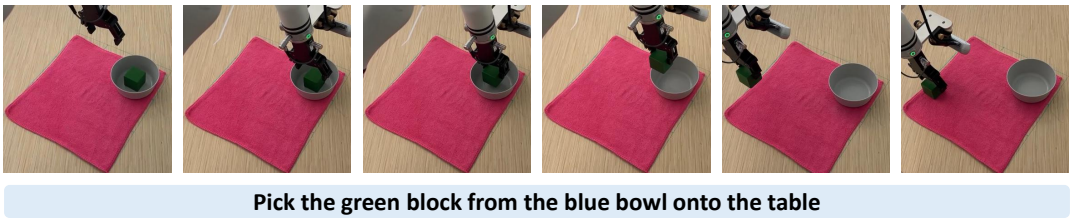
We evaluate the fine-tuned model on seven groups with 10 trials for each group. The first five groups contain the tasks the same as data collection. The last two groups are designed to evaluate the generalization ability of the models. For the "change block color" group, we repeat the previous five tasks but change the green block into blue and red ones. For the "change table cover" group, we change the table cover from red to brown and blue ones.

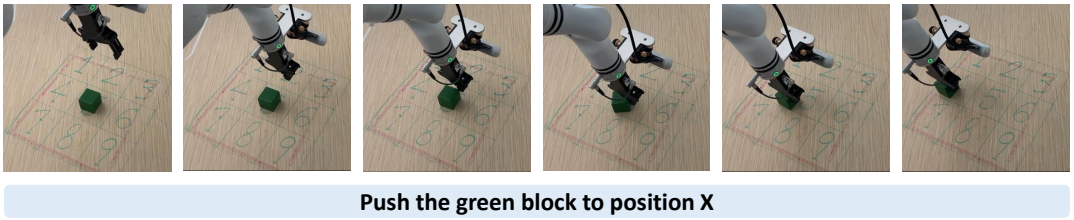
The visualization example of each task for our model can be found in Figure 10.

E.2 XHand dexterous hand

The Xhand setup is shown in Figure 8 (lower). The 12-dof Xhand is mounted on a 7-dof XArm robot arm. There are two camera views, including a main 3-rd view camera, and a wrist camera. During fine-tuning, we reinitialize linear encoder and decoder modules for both state and action to accommodate the hand's higher dimensionality.

We use the dataset collected in [26] as our finetuning dataset, which comprises roughly 4,000 trajectories spanning 13 task categories and over 50 unique objects. For evaluation, we focus on five representative XHand tasks as depicted in Figure 8, namely pick-and-place, cube stacking, upright cup placement, water pouring, and ball flicking. Each task is assessed under "seen" and "unseen" conditions: in the seen setting, the same objects and backgrounds encountered during training are used, albeit with randomized tabletop positions and optional distractors; in the unseen setting, either the target objects or the scene background (or both) were never encountered during finetuning, totaling more than 20 novel objects. During evaluation, we conducted 50 evaluation runs for the pick-and-place task, 20 runs for cube stacking, and 10 runs for each of the remaining tasks. The visualization example of each task can be found in Figure 11 and Figure 12.





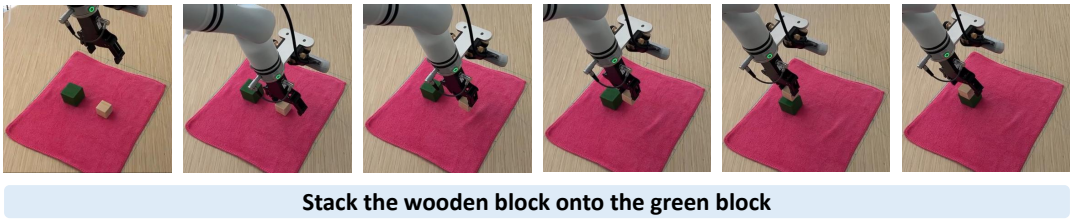


Figure 10: Realman evaluation trajectory examples.

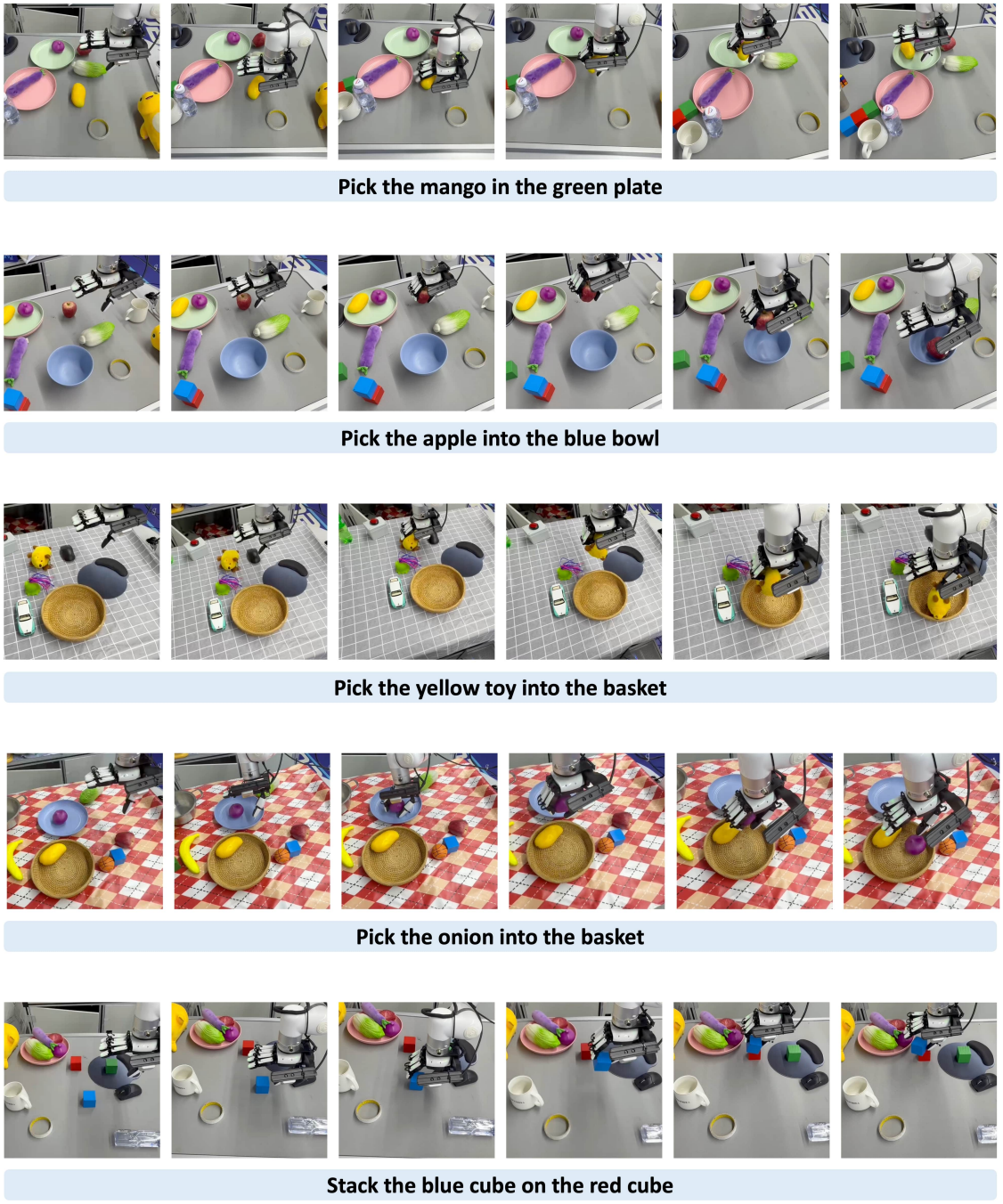
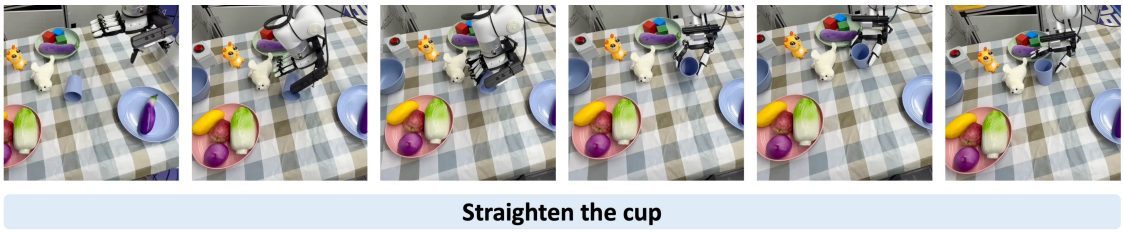
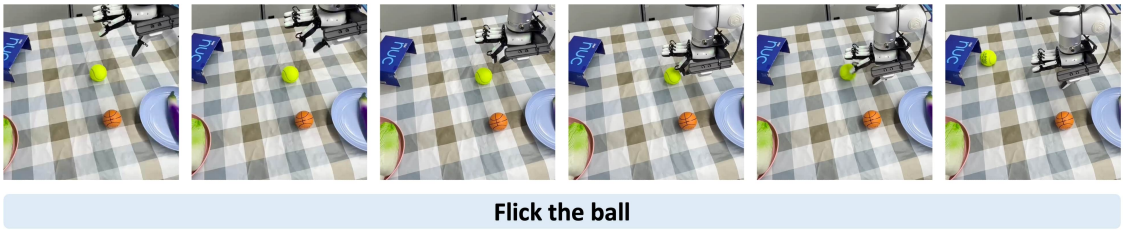


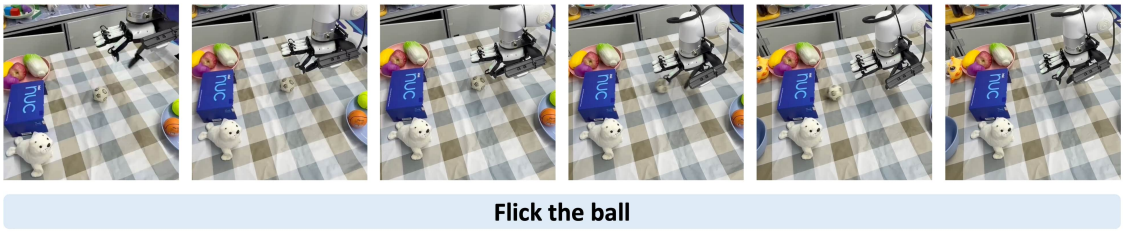
Figure 11: Xhand evaluation trajectory examples (part I).



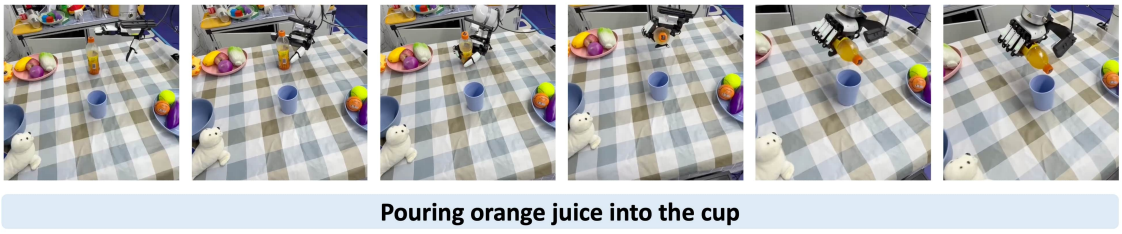
Straighten the cup



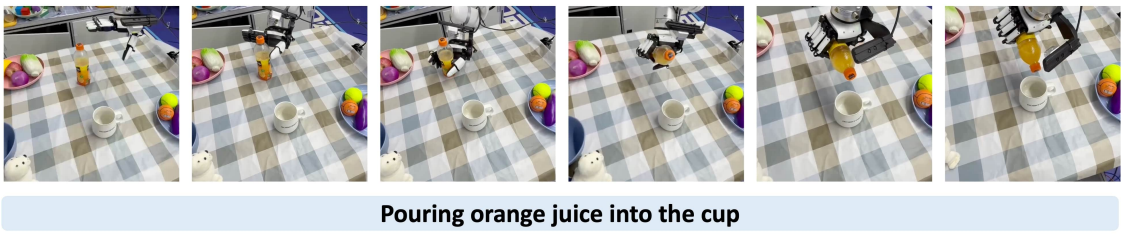
Flick the ball



Flick the ball



Pouring orange juice into the cup



Pouring orange juice into the cup

Figure 12: Xhand evaluation trajectory examples (part II).