Scalable Vision-Language-Action Model Pretraining for Robotic Manipulation with Real-Life Human Activity Videos

Qixiu Li[1,2*†] **Yu Deng**[2*] **Yaobo Liang**[2*] **Lin Luo**[2*] **Lei Zhou**[2†] **Chengtang Yao**[2]
Lingqi Zeng[2†] **Zhiyuan Feng**[1,2†] **Huizhi Liang**[1,2†] **Sicheng Xu**[2] **Yizhong Zhang**[2] **Xi Chen**[2]
Hao Chen[2] **Lily Sun**[2] **Dong Chen**[2] **Jiaolong Yang**[2‡] **Baining Guo**[2]

[1]Tsinghua University [2]Microsoft Research Asia

https://microsoft.github.io/VITRA/

Abstract

This paper presents a novel approach for pretraining robotic manipulation Vision-Language-Action (VLA) models using a large corpus of unscripted real-life video recordings of human hand activities. Treating human hand as dexterous robot end-effector, we show that "in-the-wild" egocentric human videos without any annotations can be transformed into data formats fully aligned with existing robotic V-L-A training data in terms of task granularity and labels. This is achieved by the development of a fully-automated holistic human activity analysis approach for arbitrary human hand videos. This approach can generate atomic-level hand activity segments and their language descriptions, each accompanied with framewise 3D hand motion and camera motion. We process a large volume of egocentric videos and create a hand-VLA training dataset containing 1M episodes and 26M frames. This training data covers a wide range of objects and concepts, dexterous manipulation tasks, and environment variations in real life, vastly exceeding the coverage of existing robot data. We design a dexterous hand VLA model architecture and pretrain the model on this dataset. The model exhibits strong zero-shot capabilities on completely unseen real-world observations. Additionally, fine-tuning it on a small amount of real robot action data significantly improves task success rates and generalization to novel objects in real robotic experiments. We also demonstrate the appealing scaling behavior of the model's task performance with respect to pretraining data scale. We believe this work lays a solid foundation for scalable VLA pretraining, advancing robots toward truly generalizable embodied intelligence.

1 Introduction

Pretraining on large, generic data is the key for models to acquire commonsense knowledge and achieve domain generalization. While the pretraining of Large Language Models (LLM) and Vision Language Models (VLM) has seen remarkable success [1, 13, 52, 81], pretraining Vision-Language-Action (VLA) models for dexterous hand manipulation remains largely underexplored.

Existing Vision-Language-Action data for robotic manipulation are typically collected in laboratory settings through human teleoperations [14, 28, 44, 63, 109]. Although such robot action data is invaluable, its high acquisition cost significantly limits both the scale of the collected data and its diversity in skills, object categories, and scene variations. Consequently, current V-L-A datasets lag far behind the Internet-scale language and VL data in terms of quantity and diversity, and they fall short of representing the complexity required for real-world robotic tasks. The V-L-A data for dexterous robot hands is even more scarce; to our knowledge there are no large-scale dexterous hand action datasets available for pretraining.

[*] Equal contribution. [†] Intern work done at Microsoft Research Asia. [‡] Corresponding author.

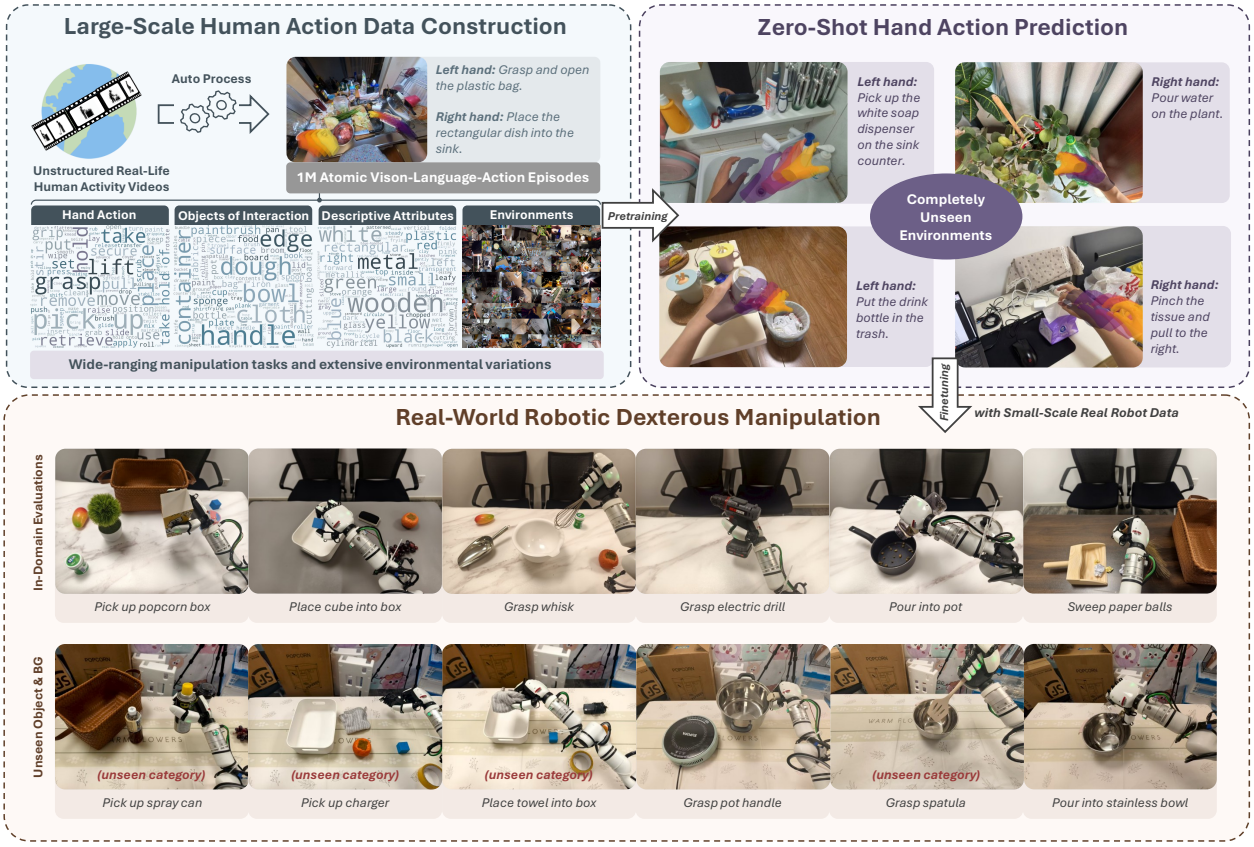


Figure 1: We present a pretraining approach for robotic Vision-Language-Action (VLA) models by transforming unstructured real-life videos of human activity into structured V-L-A formats aligned with existing robot data. The pretrained model demonstrates strong zero-shot hand action prediction in unseen environments and can be effectively fine-tuned with dexterous robot hand data for real-world tasks, showing robust generalization to new objects and environments.

Meanwhile, there is a vast amount of real-life human videos on the web, containing rich examples of everyday human actions and physical interactions with diverse environments. These videos are typically ***unstructured***: they come unscripted and unsegmented, vary in length and task granularity, contain noisy and irrelevant actions, and lack language instruction and 3D action labels. Although there have been numerous interests in utilizing human video for robot learning [4, 8, 21, 38, 56, 61, 69, 76, 84, 92, 98, 101], no existing approaches leverage large-scale, unstructured videos without any human annotation for VLA model pretraining. This leads to a critical question: ***can we transform these unstructured videos into data formats fully aligned with existing robotic V-L-A training data?***

This work is the first to address this question, and we provide an affirmative answer. For unstructured human videos, we consider human hand as robot end-effector and seek to achieve two types of alignment with real robot V-L-A data. *1) Task alignment*: we need meaningful segmentation and filtering of atomic-level human action sequences (short-horizon tasks), adhering to the recipe of existing robot data. This problem is closely related to temporal action segmentation from videos, which remains an open problem and there are no existing methods that meet our needs. *2) Label alignment*: we need to recover metric-space 3D hand motion accurately to the extent possible[1] to provide dense action labels. This is difficult as we often work with single, uncalibrated, and likely moving cameras. Additionally, we need precise language instruction labels to describe the actions.

[1]While it's ideal to have 3D motion labels as accurate as possible, we believe some noise and imperfection are acceptable for pretraining, where the goal is to grasp common knowledge, learn motion patterns for diverse skills, and experience a wide spectrum of object and scene variations.

To this end, we introduce a holistic human activity analytic framework that converts any human hand activity video of arbitrary length into multiple V-L-A trajectories of dexterous manipulation. It is a fully-automatic approach requiring no human intervention. In this framework, we first develop a monocular 3D camera and hand pose tracking approach leveraging recent advancements in 3D vision community, particularly deep visual SLAM, depth estimation, and hand reconstruction. The outputs include the camera FoV, the framewise camera pose, and the framewise hand pose (based on the 6D wrist pose and full joint angles). For temporal atomic action segmentation, we propose a simple yet surprisingly effective algorithm based on the hand movement speed in the 3D space, obtained from the recovered 3D motion labels. Finally, for each segmented video clip, we visualize hand trajectories on sampled video frames and prompt VLM to determine whether the action constitutes meaningful manipulation and, if so, describe it in natural language.

One significant advantage of real-life video data is the inherent action diversity and scene variation it offers. As a starting point, we process a large volume of raw videos from existing egocentric human video datasets. The resultant hand V-L-A dataset contains about 1 million episodes and 26 million frames, This dataset captures a broad spectrum of objects, concepts, skills, and environmental variations, vastly exceeding the coverage of existing robot data as shown in our analysis. We also develop a dexterous hand VLA model architecture with a Causal Action Transformer and pretrain the model on this dataset. The model exhibits strong zero-shot capabilities on observations of completely new scenes, a level of performance not seen in any prior method. We conduct real-world robot experiments and show that fine-tuning the model on a small amount of real robot hand data significantly improves task success rates and generalization to novel objects and backgrounds. Furthermore, our experiments show *a clear scaling behavior of task performance with respect to pretraining data scale.*

Our work stands distinct from prior research that utilizes human video for training robotic manipulation models. The approaches that leverage egocentric human video for learning vision and language representations, affordances, point trajectories, *etc.* [4, 38, 56, 61, 92, 98], did not explore action pretraining for VLA models. Recent works that use latent actions from human videos [8, 21, 101] for pretraining do not provide explicit 3D action labels as we do. Our experiments demonstrate the superiority of our pretraining approach. Most recently, a few works that are concurrent to ours studied training VLA models with explicit 3D hand motions similar to ours [7, 55, 100], but their data is largely limited to scripted laboratory captures; a detailed discussion is provided in the next section.

Our approach offers a more tractable way for pretraining data scaling compared to existing techniques. Although this work uses videos from existing egocentric video datasets, there are no technical barriers preventing further data scaling. By not imposing constraints on the subjects' activities or environments and requiring only a single webcam, every life recorder can effectively become a robot teacher. We envision a future where robots can effectively learn from abundant, low-cost human video demonstrations to acquire diverse skills, complemented by targeted fine-tuning using a modest amount of real robot data or reinforcement learning. *Our training dataset and pretrained VLA models will be open-sourced to the community to facilitate further research.*

2 Related Works

Robotic VLA Model Pretraining Robotic VLA models [9, 12, 39, 48, 49, 53, 63, 72, 83, 89, 90, 109] that can perform diverse language-instructed tasks typically need pretraining on large data. Incorporating VL-pretrained modules or backbones has been a common practice for VLA models, and here we focus on a brief overview of the pretraining with regard to *the action modality* or its proxy. Most recent VLA models with action pretraining [9, 39, 46, 48, 63, 72, 83] have leveraged the Open X-Embodiment (OXE) dataset [63], which contains over 1M real robot trajectories collected on over twenty robots. This large-scale dataset provides diverse skills and environment variations well suited for pretraining. Some of these works [9, 39, 53, 72] also incorporate more open-source or in-house robot action data in addition to OXE. The work of [27] synthesized a large volume of V-L-A data in simulators for pretraining, but it handles the grasping task only. A line of works [8, 14, 15, 21, 101] studied learning latent action from human and/or robot videos in an unsupervised manner and pretraining models using the extracted latents as the proxy for action. Some other works propose to use the future frames in videos as the prediction target for pretraining [17, 91].

There are some recent attempts concurrent to ours which use 3D hand action labels of egocentric human videos for VLA pretraining [7, 55, 100]. They primarily use hand-object interaction videos captured in controlled environments with privileged information. For example, the videos are well segmented to language-instructed action clips since the tasks are pre-scripted, and 3D hand motions are typically obtained with advanced devices (*e.g.*, RGBD sensors, VR/AR headsets). We focus on a different goal, *i.e.*, harnessing unscripted real-life human videos for large-scale pretraining, which encompass a significantly broader range of tasks, objects, and real-world environments. This greatly enhances the zero-shot action perdition performance. Furthermore, their casual capture nature facilitates much greater scalability.

Dexterous Hand Manipulation Dexterous manipulation with multi-fingered robot hands has been a vibrant area of research for decades. Earlier learning-based models with visual inputs were typically trained with reinforcement learning in simulators [2, 3, 22]. However, training dexterous RL policies requires sophisticated reward design and their applicability in real-world scenarios is often limited. Using human teleoperated demonstrations for imitation learning was also widely used to improve task performance [40, 73]. Methods that utilize human hand motion as demonstration data [34, 43, 64, 69, 71, 84, 85] were also actively studied. These previous works typically address a single or small set of tasks for a trained model. Recently, language instruction has been incorporated into dexterous hand manipulation models to handle more tasks with diverse objects [26, 38, 108].

Robot Learning from Human Videos Exploiting human videos to train robotic models has been actively studied in recent years. Several studies [56, 61, 92, 96] leverage egocentric human videos for learning vision and language representations. Some methods use explicit human actions extracted from mocap videos [23, 24, 43, 69, 71, 84] or web videos [65, 76] to guide robot policy training with imitation learning frameworks. Instead of using explicit motions, other approaches learn affordances [4, 19, 42, 57], point trajectories [6, 88, 98], or hand-object masks [78] from human videos. Recently, a group of methods have emerged which learn latent actions from human videos in an unsupervised manner and pretrain action model with latent action labels [8, 16, 21, 101, 107]. Some recent attempts use extracted 3D hand action labels from egocentric human videos for VLA pretraining [7, 55, 100]. As mentioned earlier, these primarily involve videos captured in controlled environments with privileged information. In a different vein, some approaches utilize human videos to train video generation models for human-to-robot video transfer [93, 95], visual task planning [5, 38], or world models [21, 41, 59].

Temporal Action Segmentation for Videos Temporal action segmentation, also known as temporal action detection or localization, is a technique for detecting action windows and classifying them from a long human video. Earlier approaches [29, 51, 77, 86, 104] have focused on predefined action classes. Recently, video-input VLMs [18, 20] with broad action understanding capabilities are proposed but they still face challenges in action localization accuracy. They do not meet our requirements in our preliminary tests.

3 Transforming Human Hand Video to VLA Data

Existing robotic manipulation V-L-A data [14, 28, 44, 63, 109] typically comprise simple, short-horizon tasks (*e.g., "pick up the sponge on table", "wipe the stove with cloth"*), which can be composed to long-horizon tasks by a high-level planner. Each data episode comprises a language instruction, a video frame sequence, and frame-aligned 3D action chunks of the end-effector in the robot or camera coordinate system. Our approach analyzes an unscripted human video and generates V-L-A data in such format, treating the two human hands as the end-effector. The whole framework comprises three stages and an overview is shown in Fig. 2.

3.1 3D Motion Labeling

The first stage of our approach extracts 3D motions from videos, including the motions of two hands and the camera. To achieve this, we first apply a simple algorithm to determine whether the camera is static or moving based on background optical flow. Then we estimate camera intrinsics of the videos by applying DroidCalib [33] for moving cameras and MoGe-2 [87] and DeepCalib [11] for static cameras. The videos with large distortions are then undistorted to conform to the pinhole camera model. Given the intrinsics and

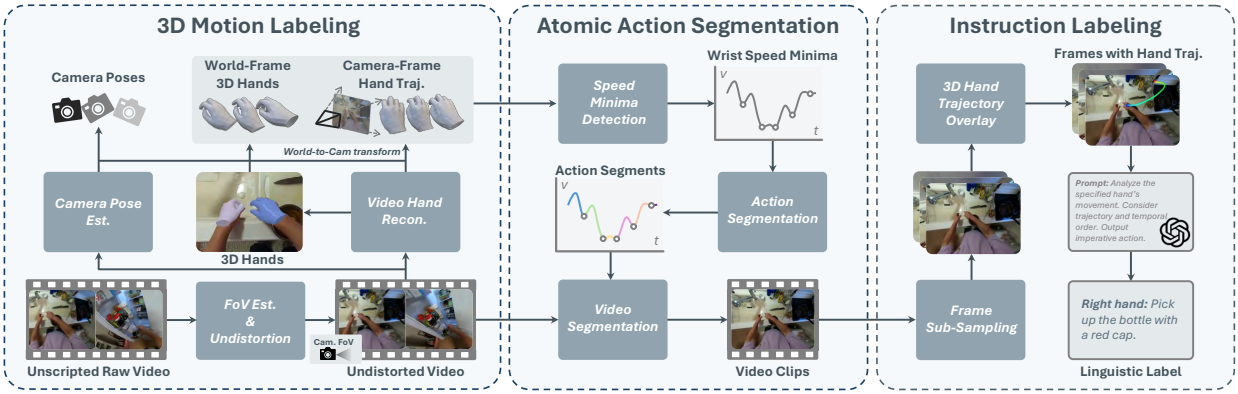


Figure 2: Overview of our holistic human activity analysis framework, which transforms unscripted real-life human videos into V-L-A episodes of human hands aligned with typical robotic data via three stages: (a) 3D motion labeling, reconstructing metric-scale 3D hand and camera trajectories; (b) atomic action segmentation, dividing videos into atomic-level clips; and (c) instruction labeling, employing GPT to annotate action instructions for each clip.

undistorted video, we proceed with video hand reconstruction and camera pose tracking. For the former, we employ HaWoR [105] to reconstruct per-frame camera-space 3D hands. Each reconstructed hand contains wrist 6D pose and joint angles represented with the MANO [75] hand parametric model. To track camera pose for moving cameras, we apply a modified version of MegaSAM [50], in which we replace the depth estimation model providing depth priors for visual SLAM with MoGe-2 [87]. Then we can obtain a sequence of world-space 3D hands by combining the camera-space 3D hands and metric-scale camera poses. Finally, we apply spline smoothing to the world-space hand motions and remove outliers. More details can be found in Appendix A.1.

The world-space 3D hand sequence can be easily transformed into any video frame's camera space, effectively simulating a static camera as in most robot data. Moreover, it facilitates both the subsequent atomic action segmentation and instruction labeling, as will be described later. To enhance efficiency, we chop long videos into overlapping 20-second clips in this stage and recompose their results.

3.2 Atomic Action Segmentation

This stage aims to segment out simple, atomic-level hand action sequences from a long video, in line with the granularity and time windows of robotic V-L-A data. This is not a straightforward task and there's no off-the-shelf models that can be applied to this problem reliably. Our solution is inspired by the natural *"beats"* of human hand action in real life. Specifically, during action transitions, human hands typically exhibit speed changes, with minima often indicating switches of action. This observation has inspired us to leverage the recovered 3D hand motions and design the following algorithm which is simple yet surprisingly effective: *we detect speed minima of the 3D hand wrists in the world space and use them as cutting points.* We smooth the hand trajectory and select points that are local speed minima within a fixed window centered on each point. Segmentation is applied for the left and right hands independently with the other hand's motion ignored. This way, each segment captures a single atomic action of at least one hand.

It is worth noting that this method is highly efficient and requires no additional model inference or pre-annotated text labels, making it particularly effective for the scalable segmentation of hand activity videos. Furthermore, segmenting atomic-level action clips can help reduce the complexity of subsequent instruction captioning, as we discuss in a later section. This strategy may lead to over segmentation for certain actions (*e.g.*, consider a wiping action where a hand moves back and forth), but these actions can be easily merged later after instruction labeling.

3.3 Instruction Labeling

Given the video segments and 3D hand action sequences, we create visualizations and utilize GPT-4.1 [1] for action captioning. From each segment, we evenly sample 8 frames and highlight hand trajectories on each frame by projecting the world-space trajectory of the hand palm from the current frame to the end of the clip (see Fig. 2 for an example). These frames are then fed into GPT, which is prompted to describe the specified hand's action in imperative form, taking into account both the content of the frames and the overlaid trajectories. We also instruct GPT to label clips lacking semantically meaningful action as "N/A". A detailed description of the prompt design can be found in the Appendix A.1.

We empirically find that providing GPT with atomic-level video clips for captioning is effective in improving annotation accuracy. By contrast, simply splitting the video into fixed-length segments (*e.g.*, 1-second) reduces accuracy, likely because each segment may still contain multiple atomic actions, which increases the difficulty for GPT to reason about the content. Additionally, overlaying hand trajectories on the images is also important for ensuring correct captioning, as evidenced by prior studies incorporating visual markers as supplementary prompts [97].

3.4 Hand V-L-A Dataset Construction

Leveraging the above framework, we construct a large-scale human hand V-L-A dataset by processing egocentric human videos from Ego4D [31], Epic-Kitchen [25], EgoExo4D [32], and Something-Something-V2 (SSV2) [30]. Note that *the human annotations for actions provided by these datasets are NOT used in this work; instead, we process the raw videos through our framework*. These annotations often do not match the desired task granularity or they lack precise start and end times for actions. Later we'll show in our experiments that training using these annotations results in obvious performance degradation compared to our approach. Our constructed dataset contains 1M episodes with 26M frames (77% from Ego4D, 12% from Epic-Kitchen, 6% from EgoExo4D, and 5% from SSV2). It features diverse hand actions, objects, attributes, and environments, encompassing real-life activities such as cooking, cleaning, construction, repairing, crafting, and painting (Fig. 1). A more detailed analysis of the dataset will be presented in Sec. 5.1.

4 Dexterous Hand VLA Model

We construct a VLA model $\boldsymbol{\pi}$ for dexterous manipulation:

$$\boldsymbol{\pi} : (\boldsymbol{l}, \boldsymbol{o}_t, \boldsymbol{s}_t) \rightarrow (\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, ..., \boldsymbol{a}_{t+N}), \tag{1}$$

which predicts a sequence of future end-effector actions $\boldsymbol{a}$ based on the current visual observation $\boldsymbol{o}_t$, the robot proprioceptive state $\boldsymbol{s}_t$, and a language instruction $\boldsymbol{l}$.

4.1 VLA Model Design

4.1.1 Model Architecture

An overview of our model architecture is presented in Fig. 3. Our model consists of a VLM backbone and a diffusion action expert. We use PaliGemma-2 [80] as the VLM, which combines a SigLIP [102] vision encoder with linear projection for alignment and a Gemma-2 [82] language model for multi-modal token processing. We use the 3B-parameter model with an input image resolution of 224^2 as the default setting. We further incorporate camera FoV information as an extra token to the model to help it better interpret the original image's aspect ratio and camera intrinsics. Following [48], we append a learnable "cognition" token as extra input to the VLM, whose output feature $\boldsymbol{f}^c$ serves as the condition for the action expert.

For the action expert, we apply a Diffusion Transformer (DiT) [67] and the DiT-Base model is used by default. The input is a concatenation of the cognition feature $\boldsymbol{f}^c$, the hand state $\boldsymbol{s}_t$, and a noisy action chunk $(\boldsymbol{a}_t^i, \boldsymbol{a}_{t+1}^i, \ldots, \boldsymbol{a}_{t+N}^i)$, where i denotes the denoising step. The hand state includes the wrist translation and rotation in camera space of the current image observation, as well as hand joint angles. A set of action masks, indicating whether each action is valid, is also provided to the model and will be discussed in detail

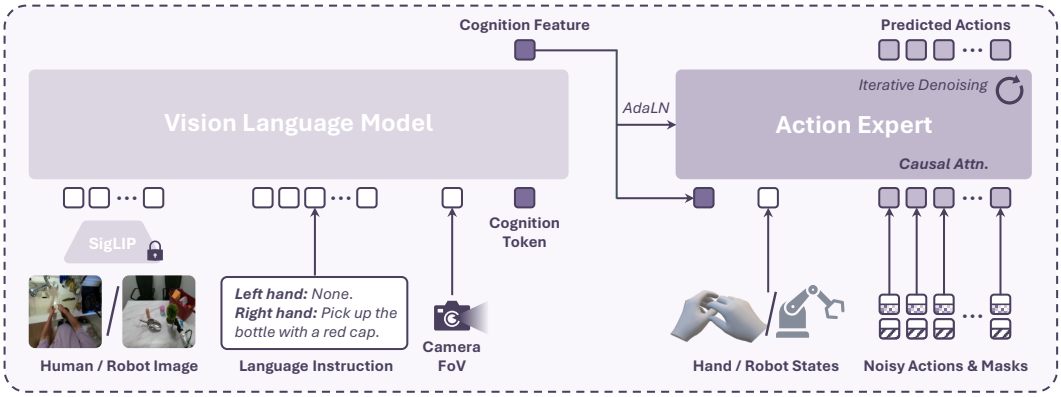


Figure 3: Our VLA model architecture. It consists of a VLM backbone and a diffusion action expert. The VLM receives visual and linguistic instructions, as well as the camera FoV, and outputs a cognition feature that guides the action expert for future action chunk prediction. The action expert additionally receives the current state of the end effector and valid action masks for iterative action denoising via causal attention.

later. We additionally inject the cognition feature into the DiT using AdaLN [67] for enhanced conditioning. The action expert predicts the added noise for iterative denoising, trained by optimizing an MSE loss:

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,1),i}||\hat{\boldsymbol{\epsilon}}^i - \boldsymbol{\epsilon}||_2, \tag{2}$$

where $\hat{\boldsymbol{\epsilon}}^i$ and $\boldsymbol{\epsilon}$ denote the predicted and ground-truth noise, respectively. The action expert, the VLM, and the cognition token are trained end-to-end, while the vision encoder remains frozen. See Appendix A.2 for more implementation details.

4.1.2 Hand Action Space

Our model predicts hand actions in the camera coordinate frame of the current observation $\boldsymbol{o}_t$. At time step t, the hand action $\boldsymbol{a}_t$ is defined as:

$$\boldsymbol{a}_t = [\Delta\boldsymbol{t}^l, \Delta\boldsymbol{r}^l, \boldsymbol{\theta}_h^l, \Delta\boldsymbol{t}^r, \Delta\boldsymbol{r}^r, \boldsymbol{\theta}_h^r] \in \mathbb{R}^{102}, \tag{3}$$

where $\Delta\boldsymbol{t} \in \mathbb{R}^3$ and $\Delta\boldsymbol{r} \in \mathbb{R}^3$ are the relative wrist translation and rotation (Euler angles converted from rotation matrices) between the consecutive frames, and $\boldsymbol{\theta}_h \in \mathbb{R}^{15\times3}$ represents the Euler angles of 15 joints in the local frame of the MANO hand model (see Fig. 8b for an illustration). Superscripts l and r indicate the left and right hand, respectively.

4.1.3 Unified Single- and Dual-Hand Action Prediction

Our VLA pretraining data is at the level of single-hand atomic actions, with some episodes containing overlapping dual-hand actions. We introduce the following designs to handle different cases in a unified manner. Specifically, the VLM always receives language instructions in the format of `Left hand: <left-hand action>. Right hand: <right-hand action>`. For a video frame $\boldsymbol{o}_t$, left- and right-hand action descriptions are set to either `None` or the instructions of the corresponding atomic action chuck $\boldsymbol{o}_t$ falls within. Meanwhile, the action expert always receives noisy hand actions for both hands. To account for episodes where action labels are available for only one hand, extra action masks (0 or 1), matching the dimensionality of the hand actions, are concatenated with the noisy actions along the feature dimension as input to the action expert (see Fig. 3). When a mask value is 0, the corresponding noisy action is set to 0 and excluded from the loss computation.

4.1.4 Causal Action Denoising

Human hands move fast in real life activities, and many of the action clips in our pretraining dataset are as short as 1 second (∼30 frames). Consequently, many prediction chunks of the VLA model go beyond the

episode end for a reasonable chunk length (*e.g.,* $N = 16$ in our setting). Naively padding with zero actions at the end can be problematic, as many atomic actions occur mid-task and should not conclude with no motion (*e.g.*, a wiping task with a hand moving back and forth). To address this issue, we employ *causal attention* for action denoising, ensuring that the token of each action step only attends to preceding actions. This prevents zero-padded positions from affecting earlier predictions, unlike in the bidirectional attention setting. Furthermore, these padded positions are also excluded from the loss computation with their corresponding action masks set to 0.

4.2 Pretraining with Human Hand VLA Data

We first train the VLA model for human hand action prediction using the dataset constructed in Sec. 3.4. During training, we apply *trajectory-aware augmentation* to the training images and actions to enhance generalization. Specifically, input images are randomly cropped and perspective-warped with varying FoV, aspect ratio, and crop center, while keeping the principal point at the image center. The action sequences are transformed accordingly to match the augmented camera parameters. During random cropping, we ensure that the projected hand trajectory from the current frame to episode end remains within the cropped image. Using this strategy, the objects of interaction are also mostly well-contained. We also apply random image flipping and make corresponding adjustments to hand actions and language instructions. Random color jittering is further applied when the text description does not contain explicit color cues.

4.3 Fine-tuning for Robotic Dexterous Manipulation

After pretraining, the model can be fine-tuned on robot data for deployment. We consider the human hand action space as a superset of that of the robot hand and align the robot's action space with the human hand's as defined in Eq. (3). Specifically, robot end-effector 6D poses in camera coordinates are used to compute $\Delta \boldsymbol{t}$ and $\Delta \boldsymbol{r}$. For joint angles, a simple mapping strategy is applied: each joint of robot hand is mapped to its closest human joint in topology, and the corresponding dimension in human action $\boldsymbol{\theta}_h$ is used for fine-tuning (an example is shown in Fig. 8b). Unmapped dimensions in $\boldsymbol{\theta}_h$ are zero-padded in the action mask. In addition, we supervise the model with direct future execution commands for the hand joints, instead of using action labels derived from the recorded robot states. This approach produces more plausible hand motions during hand-object interactions. The language instruction format is kept consistent with those used during pretraining.

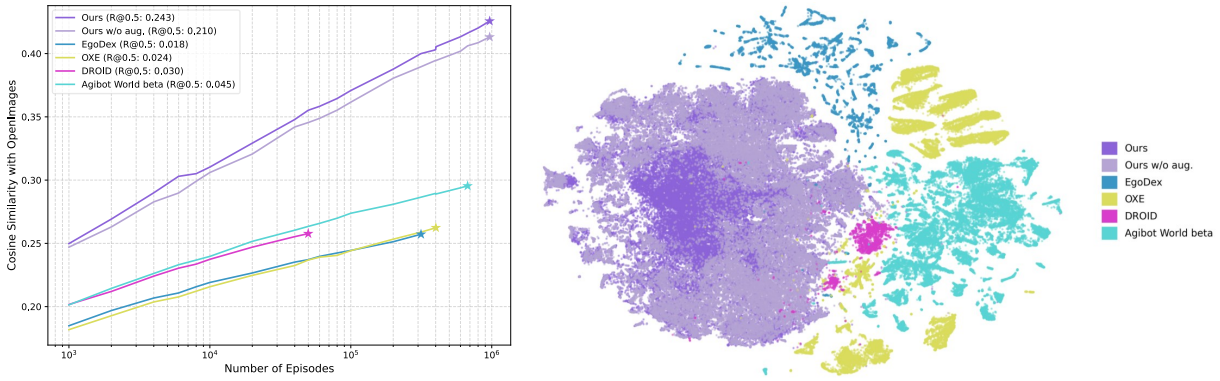
A Remark. Action space mapping between human hand and dexterous robot hand have been actively studied in the past [34, 71, 85]. In this work we do not perform direct pose transfer (as done in teleoperation) and our fine-tuning can help mitigate the action space differences. Other strategies can also be employed and we leave it as our future work.

5 Experiments

Training Details. For pretraining, we first warm up the action expert, the mapping layers of the cognition token, and the MLP projecting FoV for 5K steps. Then we jointly fine-tune the VLM backbone and action expert for 80K steps. The learning rates are 1e-4 and 1e-5 for the action expert and VLM, respectively, with a batch size of 512. The pretraining stage takes 2 days on 8 NVIDIA H100 GPUs. For fine-tuning on real robot data, we optimize the model for 20K steps with a batch size of 256 and a learning rate of 1e-5, which takes 8 hours with 8 NVIDIA H100 GPUs. More details can be found in Appendix A.3.

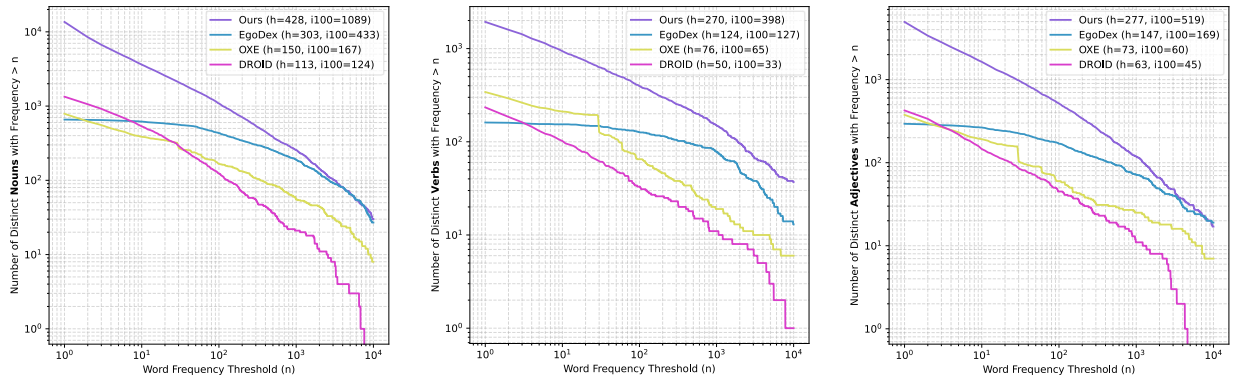
5.1 Pretraining Data Analysis

An overview of our pretraining data is shown in Fig. 1. We visualize the most frequent words in the language instructions using word clouds, and showcase randomly-sampled task environments. More examples of the dataset can be found in the Appendix C.1. To further investigate data diversity, we conduct a detailed analysis of the visual observations and language instructions in the dataset, as described below.



(a) (b)

Figure 4: Visual diversity across VLA datasets. (a) Image feature similarity with OpenImages [47] as the number of episodes varies. ⋆ marks the full dataset's similarity. (b) t-SNE visualization of image features.



(a) Nouns (b) Verbs (c) Adjectives

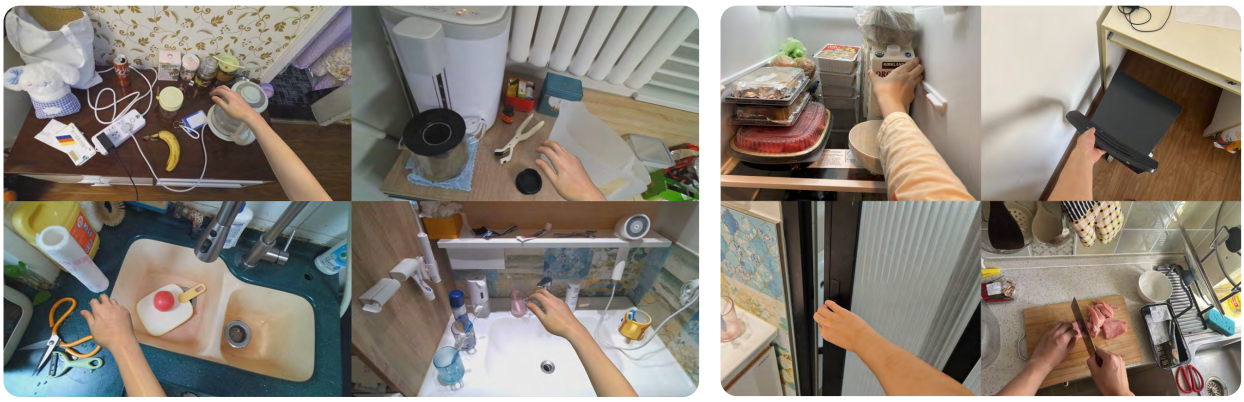
Figure 5: Language instruction statistics across different VLA datasets.

5.1.1 Visual Diversity

The diversity of visual observations and their coverage of natural scenes are crucial for enhancing the model's generalization ability in real-world scenarios. To quantify the diversity and coverage of our dataset, we use the OpenImages [47] dataset as a reference and compute the similarity between our dataset and it, as OpenImages spans a broad spectrum of real-world scenes and is known to be highly diverse [94]. Specifically, we randomly sample 8K images from OpenImages as queries and extract their features using the DINOv2 [62] encoder. For each query feature, we compute its maximum cosine similarity to our dataset, where the target features are extracted from the first frame of each episode in the dataset. We use the average of the maximum cosine similarities for all query features as a measure of dataset diversity. Higher similarity values indicate that the dataset covers a larger portion of real-world scenes represented in OpenImages.

Figure 4a presents the similarity curve as a function of the number of episodes, where we randomly sample varying numbers of episodes from the dataset and compute feature similarity as described above. We also report R@0.5, *i.e.*, the fraction of OpenImages queries with a maximum similarity above 0.5 to the target dataset features. We compare our dataset with existing VLA datasets, including EgoDex [37], a human-hand VLA dataset of over 300K episodes collected in lab environments, and widely-used robotic VLA datasets: Open X-Embodiment (OXE)[2] [63], DROID [44], and AgiBot World beta [14]. As shown in the figure, our dataset exhibits higher similarity to the OpenImages dataset, indicating greater diversity and broader coverage of real-world scenes. Even when sampling a small subset of our dataset (10K episodes), its diversity

[2]We use a subset comprising approximately 400K episodes widely used for VLA pretraining, following [45, 48].



(a) Grasping (b) General actions

Figure 6: Examples of environments used in hand action prediction evaluation.

already surpasses that of the other datasets. In addition, the diversity of visual observations can be further increased by leveraging the augmentation strategies described in Sec. 4.2. Moreover, our similarity increases more rapidly with the number of episodes (*i.e.*, with a steeper slope), indicating a more uniform coverage of real-world scenes, in contrast to the fragmented distribution observed in OXE [94]. The t-SNE visualization of image features in Fig. 4b also aligns with the observations.

5.1.2 Instruction Diversity

The diversity of language instructions is also important for the model to perform a wide range of tasks. To fairly compare datasets with varying instruction formats, we employ GPT-4.1 to extract nouns, verbs, and adjectives from each instruction and analyze their distributions separately. Figure 5 illustrates the relationship between the number of distinct words and their frequency of occurrence. A dataset with high diversity should contain a large number of distinct words, each appearing with sufficient frequency. Accordingly, curves positioned closer to the upper-right corner indicate a higher degree of instruction diversity. As illustrated, our dataset demonstrates a significantly higher degree of diversity than existing human and robot VLA datasets. Additionally, we compute the h-index and i100-index for the words, where the h-index represents the largest number h such that at least h words appear at least h times, and the i100-index counts the number of words appearing at least 100 times. Our dataset achieves higher values on both metrics compared to the other datasets.

5.2 Human Hand Action Prediction

In this section, we evaluate the performance of our pretrained VLA model for human hand action prediction in *unseen environments*. We examine how action prediction performance is influenced by key factors such as dataset composition, model architecture, training strategies, data construction strategies, and dataset scale. We first introduce the benchmark used for evaluating hand action prediction, followed by a systematic analysis of how each factor impacts performance.

5.2.1 Benchmark

We construct a benchmark under *unseen real-life environments*, consisting of two task types defined below:

Grasping We instruct the model to grasp objects in the scene. We capture RGB-D images from 47 unseen environments using Azure Kinect and annotate 396 objects with captions and segmented 3D point clouds. Synthetic human hands are rendered onto the images at distances suitable for object grasping with a single action chunk (see Fig. 6a for examples). We compute the minimum distance between predicted finger trajectories and target object points (*i.e.*, $d_{\text{hand-obj}}$) to evaluate movement plausibility.

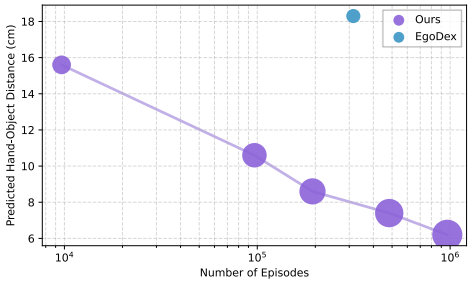
Table 1: Evaluation and ablation study of hand action prediction for the pretrained model. Note that Being-H0 [55] is a concurrent work to ours. See text for details.

Method	Grasp	General action
	Avg./med. $d_{\text{hand-obj}}$ (cm) ↓	User Score ↑
Initial position	20.0 / 20.0	–
Being-H0 (8B)	19.1 / 18.4	0.15
Ablations		
Lab data (EgoDex)	17.6 / 18.3	–
Human annotation	14.1 / 14.1	0.96
No augmentation	11.6 / 10.7	1.43
Bidirectional attention	9.3 / 7.2	1.69
Ours	**8.8 / 6.2**	**1.91**

Table 2: Ablation study on hand action prediction using different episode construction strategies. †: Pretrained using a subset of 350K episodes for efficiency.

Method	Grasp
	Avg./med. $d_{\text{hand-obj}}$ (cm) ↓
Initial position	20.0 / 20.0
Ablations	
Fixed-interval segmentation	10.5 / 8.8
No trajectory overlay	11.7 / 10.7
Ours†	**9.9 / 8.1**

Figure 7: Data scaling behavior on the grasping task. The circle size indicates the *visual diversity* of the data.



General Action For more general hand actions, quantitative metrics may not adequately capture the plausibility or correctness of the predicted movements. To address this, we design a user study to evaluate hand movements before and after contact across 117 unseen real-life environments captured with mobile phones (see Fig. 1 and Fig. 6b). For each scene, we prompt the model with annotated instructions and render predicted hand actions onto video frames. We ask 23 participants to rank the top-3 actions for 30 randomly selected scenes from the 117 environments. These actions will be assigned 3, 2, and 1 scores while all others receive 0. We then report the average scores across participants to assess the plausibility of the predicted actions.

More details of the benchmark can be found in Appendix B.

5.2.2 Performance Analysis

Comparison of Pretraining Data We first compare the performance of models trained with different pretraining datasets to validate the effectiveness of our constructed data. We compared with several baselines including *a) Lab data*, which replaces our VLA data with the EgoDex dataset captured in lab environments; *b) Human annotation*, which uses human annotations in the original human video datasets described in Sec. 3.4 for constructing VLA episodes (as mentioned previously, these annotations often do not match the desired task granularity or there's no precise start and end times); *c) Being-H0* [55], a recent hand VLA model pretrained on a large collection of scripted, laboratory human video data (as a reference).

Table 1 reports the quantitative results of different configurations on our constructed benchmark. For *grasping*, we include the initial hand–object distance as a reference. For *general action*, the *Lab data* baseline is not included because the model only predicts keypoints using labels provided by EgoDex. As shown, our method consistently outperforms all baselines. Compared to models trained on EgoDex data, ours exhibit much stronger generalizability. Being-H0, though pretrained on multiple lab datasets, still shows limited performance. Moreover, training with the original human annotations also underperforms, as its temporal

or granularity misalignment between text and actions weakens instruction following. Figure 1 further presents visual results of our model's hand action predictions in these unseen environments, demonstrating its strong generalization to diverse scenarios. More visualization results can be found in Appendix C.2.

Influence of Model Design and Training Strategy We evaluate the efficacy of model and training designs by comparing with two alternatives: *a) No augmentation*, which discards the trajectory-aware data augmentation during training; *b) Bidirectional attention*, which uses bidirectional attention for action denoising in the diffusion action expert.

As shown in Tab. 1, removing data augmentation during training largely reduces performance. This observation is consistent with the findings in Fig. 4, where data augmentation is shown to play a crucial role in enhancing the diversity of visual observations, which is essential for improving the model's generalization ability in unseen environments. In addition, replacing causal attention with bidirectional attention also leads to a performance drop, highlighting the importance of this technique which better aligns with the characteristics of our pretraining data.

Influence of Episode Construction Strategy Our framework segments unscripted human videos into atomic action episodes and generates their language instructions by leveraging *reconstructed 3D hand trajectories*. We compare our method with two baselines that omit the use of 3D hand trajectory guidance during episode construction: *a) Fixed-interval segmentation*, which segments raw videos into non-overlapping clips of 1 second instead of using speed minima as cutting points, and feeds the resulting clips to GPT for captioning; *b) No trajectory overlay*, which prompts GPT for action captioning without overlaying the projected 3D hand trajectory onto sampled frames. To improve efficiency, this ablation study is conducted on a subset of 350K episodes from Ego4D.

Table 2 demonstrates the quantitative results on grasping tasks. Using fixed-interval segmentation for constructing VLA episodes during pretraining results in degraded performance, as this approach can include multiple actions within a single clip, thereby increasing the difficulty for GPT to correctly interpret and align the corresponding action captions. Similarly, removing the hand trajectory overlay during GPT captioning also results in a noticeable performance drop. This highlights the importance of leveraging 3D hand trajectories as guidance in constructing VLA episodes.

Data Scaling Behavior We further investigate how the scale of training data influences hand action prediction performance. We compare the model trained on the full dataset with those trained on subsampled datasets at different ratios: 50%, 20%, 10%, and 1%.

Figure 7 presents the trend of performance on the grasping task with respect to the scale of the training data. The size of the circles indicate visual diversity of the training data, as defined in Sec. 5.1.1. We also plot the performance of model trained with EgoDex as a reference. As shown, the predicted hand-object distance steadily decreases with increasing data scale, following an approximately linear trend on the log scale. In addition, although EgoDex contains more episodes than our 20%, 10%, and 1% subsets, its performance still lags significantly behind the models trained on our data. We attribute this mainly to its lower data diversity, which limits its generalization ability in unseen scenarios.

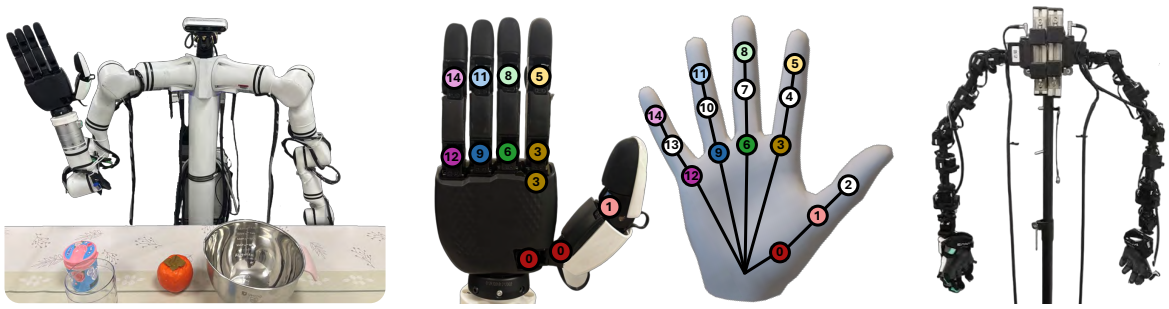
5.3 Real-World Robot Dexterous Manipulation

In this section, we evaluate the performance of our VLA model fine-tuned on a small set of real robot trajectories for dexterous manipulation tasks. We begin by describing the hardware system and the tasks defined for real-robot evaluation, followed by a detailed analysis of our model's performance and comparison with prior methods.

5.3.1 Robot Setup

We use a Realman[3] robot equipped with 12-DoF XHand[4] dexterous hands and a RealSense head camera, as shown in Fig. 8a. The robot is placed in a tabletop environment. The joint mapping between the XHand

[3]https://www.realman-robotics.com/rm75-b.html
[4]https://www.robotera.com/en/goods/2.html



(a) (b) (c)

Figure 8: (a) Robot setup in our experiments. (b) Mapping between XHand and MANO, where joints sharing the same index indicate correspondence; white color denotes joints without counterparts. (c) Teleoperation hardware system used for robot data collection.

(a) (b)

Figure 9: Fine-tuning objects/environment and unseen objects/background for real robot evaluation.

and a human hand for fine-tuning is illustrated in Fig. 8b. For real-robot data collection, we employ the teleoperation system shown in Fig. 8c. It consists of two teleoperation arms matching the Realman arms' topology for controlling end-effector 6D poses, and a pair of MANUS[5] teleoperation gloves at the arm's end for controlling the dexterous hand joint angles. See Appendix A.5 for more details of the teleoperation system.

5.3.2 Task Designs

We collected 1.2K teleoperated trajectories for four tasks: *i) General pick & place* – moving an object into a box with 3–4 random distractors; *ii) Functional grasping* – grasping an object at a functional location (*e.g.*, handle); *iii) Pouring* – picking up a bottle, pouring its contents into another container, and placing it back on the table; *iv) Sweeping* – picking up a broom from a basket, sweeping trash into a dustpan, and returning the broom. Examples of the four tasks are illustrated in Fig. 1. For evaluation, we perform the above tasks in both *seen* and *unseen* settings (see Fig. 9 for seen and unseen objects and backgrounds):

Seen Objects and backgrounds were observed during fine-tuning; randomized positions and distractors are added during evaluation.

Unseen Novel objects and backgrounds for evaluation, with two additional settings: *Unseen Objects*, where the objects are new but other objects of the same categories were seen in fine-tuning; and *Unseen Categories*, where the objects belong to categories not encountered before.

[5]https://docs.manus-meta.com/latest/Products/

Table 3: Success rates on seen real-world robot dexterous manipulation tasks (in %) .

Method	Seen Object				Average
	Pick & place (40 trials)	Functional grasp (24 trials)	Pour (8 trials)	Sweep (8 trials)	
VPP	57.5	29.2	12.5	0.0	24.8
π_0	37.5	25.0	**75.0**	50.0	46.9
No VLA pretrain	32.5	33.3	12.5	50.0	32.1
Latent action pretrain	42.5	41.7	37.5	**62.5**	46.0
OXE pretrain	40.0	37.5	62.5	25.0	41.3
Ours	**80.0**	**66.7**	**75.0**	**62.5**	**71.0**

Table 4: Success rates on unseen real-world robot dexterous manipulation tasks (in %) .

Method	Unseen Object & Background			Unseen Category & Background	Average
	Pick & place (16 trials)	Functional grasp (16 trials)	Pour (8 trials)	Pick & place (24 trials)	
VPP	12.5	0.0	0.0	8.3	5.2
π_0	0.0	6.2	25.0	33.3	16.1
No VLA pretrain	31.2	0.0	0.0	12.5	10.9
Latent action pretrain	0.0	0.0	0.0	0.0	0.0
OXE pretrain	12.5	6.3	0.0	12.5	7.8
Ours	**68.8**	**68.8**	**50.0**	**70.8**	**64.6**

5.3.3 Results and Comparisons

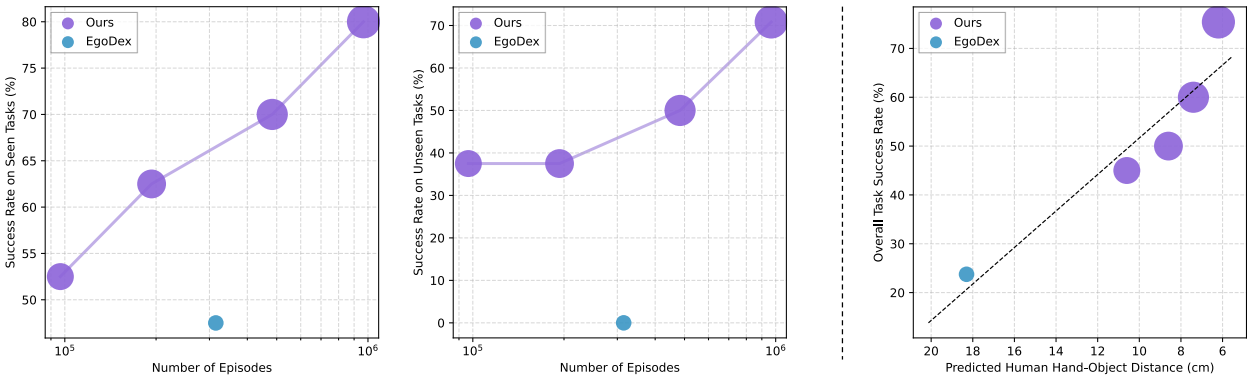
Some representative execution results of our method are presented in Fig. 1, more are presented in Appendix C.3. For quantitative evaluation, we examine the model's performance in terms of task success rate and compare with prior methods. We also analyze the effect of different pretraining data and action representations, the data scaling behavior, and the relationship between robot performance and the performance of pretraining human-hand action prediction. All baseline methods are fine-tuned using the same robot data collected in our study for a fair comparison.

Comparison with Prior Art We compare our method with two representative works: *a) VPP* [38], a recent dexterous hand manipulation model leveraging diffusion-based video generation pretraining [10]; and *b)* π_0 [9], a VLA model pretrained on extensive robot data covering a wide range of embodiments.

Table 3 and 4 compare the performance of different methods, where our approach achieves significantly higher success rates than VPP and π_0, especially on unseen tasks. In our tests, the VPP model lags behind LLM-based VLA models in instruction following and unseen object recognition; its implicit supervision through video generation pretraining seems to transfer poorly to real manipulation tasks. While π_0 is pretrained on large-scale robot data, its knowledge primarily targets gripper-based robots and does not transfer effectively to dexterous hands. Our model demonstrates robust generalization to unseen objects and environmental changes and even for objects from unseen categories, highlighting the effectiveness of leveraging human activity data for generalizable VLA learning.

Comparison of Pretraining Data To investigate the impact of different pretraining data on robot performance, we compare our method with several baselines: *a) No VLA pretrain*, which is directly fine-tuned from the base VLM model (initialized with PaliGemma-2 VLM) and a randomly-initialized action expert without human data pretraining; *b) OXE pretrain*, which uses Open X-Embodiment data instead of our human VLA data for pretraining; and *c) EgoDex pretrain*, which uses EgoDex data under lab environment for pretraining.

Results of different methods are presented in Tab. 3, 4, and Fig. 10. Compared to the model without human VLA data pretraining, our approach achieves superior execution success and stronger generalization on unseen tasks. Compared to OXE pretraining, our method achieves substantially stronger few-shot and



(a) (b) (c)

Figure 10: Data scaling behavior on real-robot pick-and-place tasks. The circle size indicates the *visual diversity* of the pretraining data. (a) Task success rate on *seen* objects and backgrounds. (b) Task success rate on *unseen* objects and backgrounds. (c) Correlation between robot task performance and pretraining hand action prediction accuracy.

unseen task performance. The OXE dataset contains data from gripper-based robots and offers far less diversity in objects, tasks, and environments compared to our human hand VLA dataset, as discussed in Sec. 5.1. Moreover, the significant differences among its various sub-datasets also reduce the model's generalization ability due to shortcut learning [94]. EgoDex pretraining leverages carefully collected large-scale human-hand data, yet its limited environmental diversity leads to lower success rates and reduced generalization performance. As shown in Fig. 10, EgoDex performs worse than our model pretrained on only 10% of the data, even though it contains more episodes and a significantly larger number of frames (*i.e.*, 130M *v.s.* 2.6M). Moreover, it completely fails on unseen scenes, highlighting the importance of data diversity for generalization.

Influence of Action Representation We further compare with a baseline method, *Latent action pretrain*, to validate the effectiveness of pretraining with explicit 3D action predictions. For this baseline, we use the same episode splits and language instructions, but replace the original 3D actions to be predicted with latent actions from LAPA [101] trained on our data.

The comparison results are also presented in Tab. 3 and 4. As shown, while latent action pretraining performs moderately on seen tasks, it fails completely in unseen environments. This is likely due to latent actions struggling to disentangle task-relevant motions from task-irrelevant background, causing them to fail in novel settings[6]. By contrast, our approach achieves significantly better performance, benefiting from more explicit action supervision, which leads to a smaller pretraining–finetuning gap. This demonstrates the advantage of closely aligning human video data with robotic VLA data during pretraining.

Data Scaling Behavior Following the human-hand prediction experiments, we examine how the scale of pretraining data affects robotic task performance. For this experiment, we compare with models pretrained on human-hand data using 50%, 20%, and 10% of the dataset (we do not include the 1% case, as we consider this amount too small for effective pretraining before fine-tuning). We focus on general pick-and-place tasks involving both seen and unseen objects (categories) and backgrounds. Figure 10 illustrates the relationship between the scale of pretraining data and robotic task success rates, showing consistent improvements on both seen and unseen tasks as the data scale increases.

Robot Performance *v.s.* Pretraining Hand-Prediction Accuracy Finally, we investigate the relationship between the fine-tuned robotic task success rates and the pretraining accuracy on human-hand prediction. The former is evaluated by the average success rate across seen and unseen pick-and-place tasks,

[6]Some recent approaches [16] focus on better disentangling task-irrelevant information in latent actions. We leave a comparison with these methods for future work.

while the latter is measured by the hand–object distance reported in Tab. 1. As shown in Fig. 10c, the two exhibit a clear positive correlation: models that achieve higher performance on the human-hand prediction benchmark also yield higher task success rates after fine-tuning on robot data. This suggests that our hand action prediction benchmark can serve as an effective proxy for downstream robotic performance, enabling rapid prototyping of pretrained VLA models.

6 Discussion and Future Work

Our method serves as an initial exploration toward constructing large-scale VLA pretraining data from real-life human activity videos. While our current data are mainly sourced from existing egocentric human video datasets, the automatic data pipeline is readily extensible. In future work, we plan to incorporate more diverse video sources (*e.g.*, Howto100M [60]) to enable larger-scale and more comprehensive VLA pretraining. Due to the limitations of current 3D reconstruction algorithms and the inherent capabilities of the VLM, our constructed pretraining data still exhibits some inaccuracies. We aim to further improve its quality using more advanced reconstruction techniques, and introduces additional filtering mechanisms to remove noisy samples. In addition, our present data construction and model training mainly target short-horizon, atomic manipulation skills. Extending this framework to organize data into higher-level task structures for learning long-horizon planning and reasoning abilities represents an important future direction.

Our current robotic experiments primarily focus on single-handed manipulation tasks. Nevertheless, our framework naturally supports bimanual operations. We conduct a simple "hand-over" experiment to demonstrate its feasibility on two-handed tasks, as shown in the bottom row of Fig. IX. We plan to investigate more bimanual scenarios in future work. Furthermore, integrating our pretraining framework with multi-view visual inputs and tactile feedback to handle more complex manipulation tasks is another promising direction for exploration.

7 Conclusion

This paper introduces a novel approach for pretraining robotic manipulation VLA models using unstructured real-life human activity videos. We develop a fully-automatic pipeline to convert in-the-wild egocentric human videos into atomic-level VLA data aligned with existing robotic demonstrations. We also design a dexterous hand VLA model with tailored training strategies to effectively leverage human data for pretraining. Experiments show that our pretrained model exhibits strong zero-shot performance in unseen real-world environments, high task success after being finetuned on limited robot data, and favorable data scaling behavior, demonstrating a highly promising and scalable approach toward learning truly generalizable embodied robots.

Acknowledgments

We would like to thank Mozheng Liao, Guanghao Wang, and Bo Liang for their help in building the hardware system, Lidong Zhou for the suggestions on analysis experiments, and Yunze Liu, Ruicheng Wang, Fangyun Wei, Yichao Shen, and Jianmin Bao for discussions on improving this work.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv:1910.07113*, 2019.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *IJRR*, 2020.

[4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, 2023.

[5] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2Act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv:2409.16283*, 2024.

[6] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024.

[7] Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-RDT: Human manipulation enhanced bimanual robotic manipulation. *arXiv:2507.23523*, 2025.

[8] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv:2503.14734*, 2025.

[9] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*, 2024.

[10] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127*, 2023.

[11] Oleksandr Bogdan, Viktor Eckstein, Francois Rameau, and Jean-Charles Bazin. DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *CVMP*, 2018.

[12] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv:2212.06817*, 2022.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[14] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IROS*, 2025.

[15] 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. *arXiv:2502.14420*, 2025.

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

[17] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv:2410.06158*, 2024.

[18] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, et al. Videollm: Modeling video sequence with large language models. *arXiv:2305.13292*, 2023.

[19] Hanzhi Chen, Boyang Sxun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *CVPR*, 2025.

[20] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-Online: Online video large language model for streaming video. In *CVPR*, 2024.

[21] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv:2411.00785*, 2024.

[22] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *NeurIPS*, 2022.

[23] Zerui Chen, Shizhe Chen, Arlaud Etienne, Ivan Laptev, and Cordelia Schmid. ViViDex: Learning vision-based dexterous manipulation from human videos. In *ICRA*, 2025.

[24] Zoey Qiuyu Chen, Karl Van Wyk, Yu-Wei Chao, Wei Yang, Arsalan Mousavian, Abhishek Gupta, and Dieter Fox. DexTransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations. *arXiv:2209.14284*, 2022.

[25] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, et al. The EPIC-KITCHENS dataset: Collection, challenges and baselines. *TPAMI*, 2020.

[26] Vincent de Bakker, Joey Hejna, Tyler Ga Wei Lum, Onur Celik, Aleksandar Taranovic, Denis Blessing, Gerhard Neumann, Jeannette Bohg, and Dorsa Sadigh. Scaffolding dexterous manipulation with vision-language models. *arXiv:2506.19212*, 2025.

[27] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, et al. GraspVLA: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv:2505.03233*, 2025.

[28] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In *ICRA*, 2024.

[29] Yazan Abu Farha and Jurgen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019.

[30] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[31] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[32] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024.

[33] Annika Hagemann, Moritz Knorr, and Christoph Stiller. Deep geometry-aware camera self-calibration from video. In *ICCV*, 2023.

[34] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020.

[35] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv:2010.04245*, 2020.

[36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[37] Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. *arXiv:2505.11709*, 2025.

[38] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *ICML*, 2025.

[39] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv:2504.16054*, 2025.

[40] Divye Jain, Andrew Li, Shivam Singhal, Aravind Rajeswaran, Vikash Kumar, and Emanuel Todorov. Learning deep visuomotor policies for dexterous hand manipulation. In *ICRA*, 2019.

[41] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, et al. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv:2505.12705*, 2025.

[42] Aditya Kannan, Kenneth Shaw, Shikhar Bahl, Pragna Mannam, and Deepak Pathak. DEFT: Dexterous fine-tuning for hand policies. In *CoRL*, 2023.

[43] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *ICRA*, 2025.

[44] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv:2403.12945*, 2024.

[45] 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:2406.09246*, 2024.

[46] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv:2502.19645*, 2025.

[47] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[48] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv:2411.19650*, 2024.

[49] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *ICLR*, 2022.

[50] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In *CVPR*, 2025.

[51] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *ICCV*, 2023.

[52] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.

[53] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. *ICLR*, 2025.

[54] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[55] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: vision-language-action pretraining from large-scale human videos. *arXiv:2507.15597*, 2025.

[56] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *ICLR*, 2023.

[57] Priyanka Mandikal and Kristen Grauman. DexVIP: Learning dexterous grasping with human hand pose priors from video. In *CoRL*, 2022.

[58] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *ICRA*, 2007.

[59] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In *RSS*, 2023.

[60] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *CVPR*, 2019.

[61] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *CoRL*, 2023.

[62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2024.

[63] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *ICRA*, 2024.

[64] Sungjae Park, Seungho Lee, Mingi Choi, Jiye Lee, Jeonghwan Kim, Jisoo Kim, and Hanbyul Joo. Learning to transfer human hand skills for robot manipulations. *arXiv:2501.04169*, 2025.

[65] Austin Patel, Andrew Wang, Ilija Radosavovic, and Jitendra Malik. Learning to imitate object interactions from internet videos. *arXiv:2211.13225*, 2022.

[66] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019.

[67] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[68] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, pages 10106–10116, 2024.

[69] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation learning for dexterous manipulation from human videos. In *ECCV*, 2022.

[70] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv:2307.04577*, 2023.

[71] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv:2503.13441*, 2025.

[72] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. *arXiv:2501.15830*, 2025.

[73] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *RSS*, 2018.

[74] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv:2408.00714*, 2024.

[75] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: modeling and capturing hands and bodies together. *TOG*, 2017.

[76] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. VideoDex: Learning dexterity from internet videos. In *CoRL*, 2023.

[77] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.

[78] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *ICRA*, 2025.

[79] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020.

[80] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv:2412.03555*, 2024.

[81] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[82] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, et al. Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*, 2024.

[83] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv:2405.12213*, 2024.

[84] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-horizon imitation learning by watching human play. In *CoRL*, 2023.

[85] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and Karen Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. In *RSS Workshop*, 2024.

[86] Limin Wang, Yu Qiao, Xiaoou Tang, et al. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 2014.

[87] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *arXiv:2507.02546*, 2025.

[88] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv:2401.00025*, 2023.

[89] 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 *CoRL*, 2025.

[90] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. TinyVLA: Towards fast, data-efficient vision-language-action models for robotic manipulation. *RA-L*, 2025.

[91] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv:2312.13139*, 2023.

[92] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv:2203.06173*, 2022.

[93] Sicheng Xie, Haidong Cao, Zejia Weng, Zhen Xing, Haoran Chen, Shiwei Shen, Jiaqi Leng, Zuxuan Wu, and Yu-Gang Jiang. Human2Robot: Learning robot actions from paired human-robot videos. *arXiv:2502.16587*, 2025.

[94] Youguang Xing, Xu Luo, Junlin Xie, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. In *Conference on Robot Learning*. PMLR, 2025.

[95] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *IROS*, 2021.

[96] Jiange Yang, Bei Liu, Jianlong Fu, Bocheng Pan, Gangshan Wu, and Limin Wang. Spatiotemporal predictive pre-training for robotic motor control. *arXiv:2403.05304*, 2024.

[97] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv:2310.11441*, 2023.

[98] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal AI agents. In *CVPR*, 2025.

[99] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pages 10371–10381, 2024.

[100] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv:2507.12440*, 2025.

[101] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *ICLR*, 2025.

[102] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[103] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *NeurIPS*, 32, 2019.

[104] Chen-Lin Zhang, Jianxin Wu, and Yin Li. ActionFormer: Localizing moments of actions with transformers. In *ECCV*, 2022.

[105] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *CVPR*, 2025.

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

[107] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. FLARE: Robot learning with implicit world modeling. *arXiv:2505.15659*, 2025.

[108] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Num Lui, et al. DexGraspVLA: A vision-language-action framework towards general dexterous grasping. *arXiv:2502.20900*, 2025.

[109] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

A More Implementation Details

A.1 Hand V-L-A Data Construction

A.1.1 3D Motion Labeling

Camera Intrinsics Estimation For videos captured with moving cameras, we apply DroidCalib to estimate the camera intrinsics under the unified camera model [58], which extends the conventional pinhole model by introducing an additional distortion parameter to better handle ultra-wide-angle and fisheye cameras. During estimation, we assume the principal point lies at the image center and the focal lengths are identical in both image axes. For static cameras, we first employ DeepCalib to estimate intrinsics under the same unified camera model assumption. If the distortion coefficients are small, we apply MoGe-2 to estimate the final focal length under the pinhole camera model assumption. For videos with non-negligible distortion, an additional undistortion step is applied to make the images conform to the pinhole model before subsequent processing.

Video Hand Reconstruction We employ HaWoR for camera-space 3D hand reconstruction which is capable of jointly reconstructing hands within each video chunk. The camera focal information estimated in the previous stage is provided as input to HaWoR to facilitate accurate hand reconstruction. The original HaWoR includes a hand motion infiller module that interpolates missing frames in the 3D reconstruction. We discard this module in our pipeline, as its interpolation results are less reliable on in-the-wild videos.

Camera Pose Estimation We use MegaSAM for metric-scale camera pose estimation. The original MegaSAM incorporates depth priors predicted by DepthAnything [99] and UniDepth [68] to address challenges in videos with limited camera baselines and complex scene dynamics. In our initial exploration, we found that replacing these depth modules with direct outputs from MoGe-2 yields more accurate and stable results, while significantly improving inference efficiency. Therefore, we adopt this modified version of MegaSAM in our framework. Additionally, when estimating the camera, we initialize MegaSAM's camera intrinsics using the focal length information obtained from the previous stage.

A.1.2 Atomic Action Segmentation

We segment long videos into atomic-level video clips using speed minima of the 3D hand wrists in the world space. Before computing the minima, the 3D wrist trajectories are smoothed with a Gaussian filter in world space to mitigate spurious extrema caused by reconstruction noise. Additionally, a detected minimum is required to be the smallest value within a 0.5-seconds window centered on it to further mitigate noise effects.

A.1.3 Instruction Labeling

We prompt GPT-4.1 to generate action captions based on sampled frames from atomic-level video clips. A detailed example is illustrated in Fig. I. After obtaining a caption for a video clip, we further ask GPT-4.1 to rephrase it into five diverse versions while preserving the original meaning, in order to increase the diversity of language descriptions.

A.2 Model Architecture

A.2.1 VLM Backbone

Our VLM is based on the 3B version of PaliGemma-2 with "mix checkpoint" that has been further fine-tuned on multiple downstream tasks. The input image resolution is 224^2, and all images of varying sizes are directly resized to this resolution without any additional center cropping. To improve the model's understanding of the original images' aspect ratio and camera intrinsics, we incorporate the camera FoV as an additional token. The FoV token is projected via an MLP to align with the embedding space of the LLM input tokens, allowing the model to interpret the geometric characteristics of visual inputs more effectively. This is important because the SigLIP encoder processes images with a fixed 1:1 aspect ratio, which may lead to

ambiguity if FoV cues are absent. Moreover, our trajectory-aware augmentation (see Sec. 4.2) modifies the FoV, necessitating the model to consider these variations for correct interpretation of the augmented images.

A.2.2 Diffusion Action Expert

The action expert is implemented as a Diffusion Transformer with approximately 136M parameters. Within each transformer block, we replace bidirectional self-attention with causal self-attention for action tokens (see Sec. 4.1.4 for further discussion), and QKNorm [35] is applied in the attention layers while LayerNorm is replaced with RMSNorm [103] to improve training stability. The cognition feature $\boldsymbol{f}^c$, the hand state $\boldsymbol{s}_t$, and the noisy action chunk are first projected via an MLP and subsequently processed through a causal self-attention layer. Additionally, $\boldsymbol{f}^c$ is injected via AdaLN to further enhance vision–language conditioning.

A.2.3 State and Action Normalization

For both state and action inputs to the action expert, we apply mean–variance normalization to each dimension, standardizing them to zero mean and unit variance. During pretraining, dataset-specific statistics are first computed, after which unified normalization parameters are derived by weighting each dataset according to its frame sampling probability. These parameters are then kept fixed throughout pretraining and zero-shot hand action prediction evaluation. During fine-tuning, we recompute the mean and variance from the robot data and apply normalization accordingly.

A.3 Training Details

During training, we employ PyTorch's Fully Sharded Data Parallel (FSDP) framework. The length of the action chunk is set to 16, and the diffusion process uses 100 noise steps. For each VLM forward pass, eight noisy samples are randomly drawn to train the action expert efficiently, following [48]. AdamW [54] is used as the optimizer with a weight decay of 1e-1 and a gradient clipping value of 1.0, applied consistently in both pretraining and fine-tuning stages. The β parameters of AdamW are set to $(\beta_1, \beta_2) = (0.9, 0.99)$ during pretraining and $(\beta_1, \beta_2) = (0.9, 0.95)$ during fine-tuning to enhance training stability.

At the pretraining stage, we employ trajectory-aware augmentation as discussed in Sec. 4.2. For episodes annotated with multiple language instructions by GPT (see Appendix A.1.3), a single instruction is randomly selected per trajectory. The state input $\boldsymbol{s}_t$ to the action expert is dropped with a probability of 0.1, encouraging the model to rely solely on vision–language input and preventing overfitting to the state. Similarly, the cognition token is dropped with a probability of 0.1 in the action expert to leverage classifier-free guidance (CFG) [36].

During finetuning, we align the coordinate systems of the real-robot and pretrained human data, and map the robot hand's action dimensions to match those of the human hand, following the discussion in Sec 4.3. By default, the pretrained model is finetuned for 20K steps. The model without VLA pretraining are finetuned for 60K steps, as they do not converge within 20K steps and produce highly jittery actions. For other configurations, including ablations and prior methods, we select the best-performing checkpoint every 10K steps. All prior methods used for comparison are implemented using their official repositories.

A.4 Inference Details

During inference, we use DDIM [79] with 10 sampling steps and a CFG scale of 5.0. For real-robot execution, we adopt the action chunking strategy [48, 106], executing 6 out of 16 actions at a time. Predicted end-effector actions in the camera coordinate frame are first converted to absolute 6D poses in the robot coordinate frame, then transformed into joint angles using an inverse kinematics (IK) solver. The resulting hand joint angles are directly mapped to the corresponding robot dexterous hand joints for execution. Experiments for *Human Hand Action Prediction* (Sec. 5.2) are conducted on a single NVIDIA A6000 GPU, while *Real-World Robot Dexterous Manipulation* experiments (Sec. 5.3) use a single NVIDIA 4090 GPU.

A.5 Robot Teleoperation System

A.5.1 Leader–Follower Teleoperation Arm

We use a leader–follower arm system for teleoperation data collection. The leader arms share the same joint topology as the Realman robot arms. Operators manipulate the leader arms to perform desired motions, and the joint angles measured from the leader arms are directly sent to the follower arms (*i.e.*, robot arms) to replicate these motions, enabling precise control of the end-effector 6D pose.

A.5.2 Hand Pose Retargeting

We use MANUS gloves to directly control the dexterous robot hand and employ a hand pose retargeting method to map the glove measurements to the joint angles of the robot hand for precise motion control. At each timestep, the retargeting algorithm optimizes the mapping to transform human hand motions into the corresponding robot hand movements. We implement two optimizers in our pipeline, as described below:

DexPilot Optimizer The first follows the approach of DexPilot [34] and the implementation of AnyTeleop [70]. We define a set of vectors $\mathcal{V}$ consisting of the five wrist-to-fingertip vectors and ten inter-finger vectors. The objective is to minimize the squared difference between the glove keypoint vectors v_i^h and the corresponding robot vectors $v_i^r(q_t)$ obtained through forward kinematics:

$$\mathcal{L}_{\text{vec}}(q_t) = \sum_{i=0}^{N} s(d_i) \left\| \alpha v_i^h - v_i^r(q_t) \right\|^2 + \beta \left\| q_t - q_{t-1} \right\|^2, \tag{I}$$

subject to $q_l \leq q_t \leq q_u$, where q_l and q_u denote the joint limits of the robot hand, α is a scaling factor to account for different hand sizes, and β is a weight for temporal smoothness. The switching weight function $s(d_i)$ increases as the distance d_i between the fingertip and wrist decreases, encouraging fingertip contact.

Angle Matching The second optimizer focuses on two key vectors: the thumb–wrist vector and the vector connecting the index fingertip to its root. Their optimization follows the same formulation as Eq. (I). Notably, only the lateral-swing degrees of freedom (*i.e.*, abduction and adduction) of the thumb and index finger are updated based on this optimization, while their flexion degrees of freedom are excluded. All remaining joints are directly controlled using angles derived from the glove keypoints.

Given glove keypoints $\mathbf{k}_i \in \mathbb{R}^3$, joint angles are computed from triplets (A_j, B_j, C_j) as

$$\theta_j^h = \arccos\left(\frac{(\mathbf{k}_{A_j} - \mathbf{k}_{B_j})^\top (\mathbf{k}_{C_j} - \mathbf{k}_{B_j})}{\|\mathbf{k}_{A_j} - \mathbf{k}_{B_j}\| \, \|\mathbf{k}_{C_j} - \mathbf{k}_{B_j}\|} \right),$$

with the sign determined by the cross product relative to a reference axis, where (A_j, B_j, C_j) denote the indices of three glove keypoints forming two connected bone segments. Each glove angle θ_j^h is then linearly mapped from its empirical range $[\theta_j^{h,\min}, \theta_j^{h,\max}]$ to the corresponding robot joint limits $[\theta_j^{r,\min}, \theta_j^{r,\max}]$:

$$\theta_j^r = \frac{\theta_j^h - \theta_j^{h,\min}}{\theta_j^{h,\max} - \theta_j^{h,\min}} \left(\theta_j^{r,\max} - \theta_j^{r,\min}\right) + \theta_j^{r,\min},$$

and clipped to the valid range $[\theta_j^{r,\min}, \theta_j^{r,\max}]$. The resulting θ^r values are applied directly as angle commands to the robot hand.

B More Evaluation Details

B.1 Hand Action Prediction Benchmark

We use the benchmark described in Sec. 5.2.1 to quantitatively evaluate the performance of the pretrained VLA models on the grasping task. The RGB-D images captured with the Azure Kinect are first undistorted

to remove camera distortion. Then, for each image, we manually annotate the object positions and their corresponding captions. The annotated object positions are subsequently used as prompts for SAM-2 [74] to obtain the corresponding object masks. These masks, together with the depth images and the camera intrinsics, are used to reconstruct the 3D point cloud of each object in the camera coordinate frame.

For each image-caption pair corresponding to a target object in the scene, we render a synthetic hand with an attached arm using SMPL-X [66], a parametric full-body model that incorporates the MANO hand. We assign the hand a natural resting pose and place it approximately 20 cm away from the target object. We ensure that the synthetic hand is always positioned closer to the camera than the target object, preventing incorrect occlusion relationships in the rendered images. The images with the rendered hand, along with the corresponding language instructions, are then used to evaluate VLA hand action prediction. We prompt the model to generate a single action chunk, which corresponds to approximately 0.5 s of motion for our model and 1 s for Being-H0, and compute the minimal distance between the fingertip positions and the object's point cloud. For each instruction, we independently generate four trajectories and report the mean of their minimal hand–object distances. The final score is obtained by averaging (or taking the median of) these distances across all image–instruction pairs.

B.2 User Study

For more general hand actions, we assess the plausibility of the action predictions through a user study. We developed a website to allow participants to evaluate the quality of actions generated by different methods. The user interface is shown in Fig. II. Each participant was assigned 30 trials, randomly selected from a total of 117 scenes. For each case, the results from different methods were anonymized and presented in a randomized order to avoid any bias. The hand motions generated by each method were rendered as videos, and participants were asked to judge whether the generated video matched the given language instruction and to rank the different methods accordingly.

C More Results

C.1 Human Hand V-L-A Data

We showcase additional hand VLA data constructed using our method in Figs. III and IV. Our dataset covers a wide range of environments and hand actions. Moreover, due to the presence of moving cameras in our scenes, the observations across different frames show noticeable variation, further increasing the diversity of the data.

C.2 Hand Action Prediction Results

We provide more hand action prediction results on *unseen real-life environments* in Fig. V and VI. Our pretrained VLA model demonstrates strong generalization across these scenes and is capable of predicting diverse human hand motions.

C.3 Real-Robot Execution Results

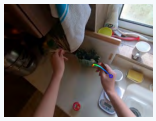
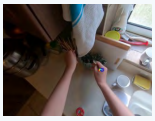
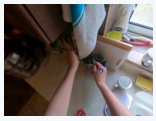
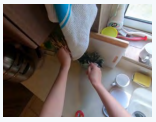
Figures VII, VIII, and IX present additional visual results of real robot task executions in our experiments. In the last row of Fig. IX, we additionally showcase a bimanual "hand over" task, demonstrating the framework's ability to transfer naturally to bimanual tasks.

```
system_prompt = """
You are a multimodal expert specializing in video captioning for egocentric human-object interaction (HOI) clips.
"""
```

```
user_prompt = """
     I will send you a set of video frames. Your goal is to describe the **specific {hand_type}-hand action** shown in
the provided video frames below. These frames are sampled from an egocentric video and contain a single atomic hand-
object interaction. A projected 2D hand trajectory is overlaid — this path represents the 3D palm center over time, with
color gradually transitioning from blue to green to red to indicate temporal progression. The {hand_type}-hand palm
position is marked with a blue dot. **Do not confuse it with the {opposite_hand_type} hand.** Respect the temporal order
of frames. Each one is labeled by number (e.g., “Frame 1”, “Frame 2”, etc.), indicating its place in the time sequence.
Please analyze the action step by step. Consider the hand status in each frame, whether there is an interacted object in
each frame, and the temporal order of the frames.

    Generate a one-sentence description of the {hand_type}-hand action shown in the entire sequence. When describing the
{hand_type}-hand action, please follow these rules:
    - **Only describe {hand_type}-hand actions**. Ignore the {opposite_hand_type} hand completely.
    - **Write in imperative form** (e.g., “Insert the key,” not “The hand is inserting…”). Do not use personal pronouns.
    - Use **specific, descriptive verbs**. If the action clearly involves picking up or placing an object, prefer verbs
like “pick” and “place” to highlight the action intent. Avoid vague or generic terms like “clean”, “spray”, or "fix".
    - Describe the interacted object only if:
        (1) the {hand_type} hand clearly interacts with it,
        (2) or, if not, the hand is purposefully moving toward it with clear intent.
        (3) If neither applies, return "N/A".
    - Be careful not to misidentify objects or their colors due to the trajectory overlay.
    - **Do not hallucinate**: if no clear or meaningful hand action, or object is present, return: "N/A".
    - Do not guess the action based on context. For example, do not assume someone is brushing something just because
there's a sink.
    - Return your answer in **JSON format** with two fields:
        (1) "think": a brief, step-by-step reasoning process (no longer than 3-4 sentences) explaining how the
{hand_type}-hand action was determined from the hand motion trajectory and visual content.
        (2) "action": the final one-sentence description of the {hand_type}-hand action, following all the rules above.

    Please prepare to receive the frames.

    Frame 1:  Frame 2:  Frame 3:  Frame 4: 
    Frame 5:  Frame 6:  Frame 7:  Frame 8: 

    Please now analyze and generate the results.
"""
```

```
response = """
{
  "think": "In Frame 1, the right hand is holding a spoon and moving toward a container of utensils. In Frames 2-4, the
hand approaches the container, and by Frames 5-8, the hand is clearly placing the spoon into the container. The hand
trajectory shows a continuous movement from the sink area to the utensil container, with the spoon always in the right
hand and being deposited among other utensils. The action is specific and involves placing the spoon into the
container.",
  "action": "Place the spoon into the utensil container."
}
"""
```

Figure I: An example of our instruction labeling process.

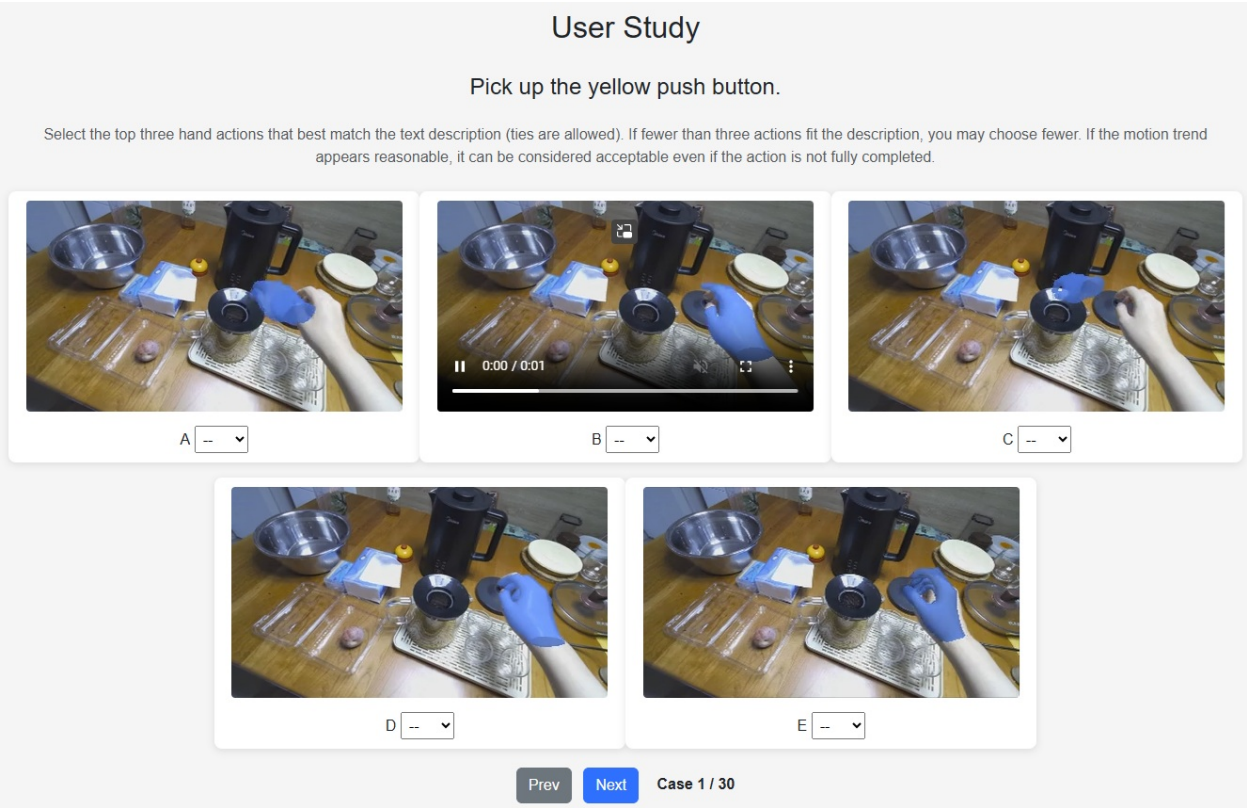


Figure II: User study interface for hand action prediction.

Right hand: *Pick up the pot lid by its handle.*

Left hand: *Push the mop forward across the floor.*

Right hand: *Wipe the stove surface with a blue cloth.*

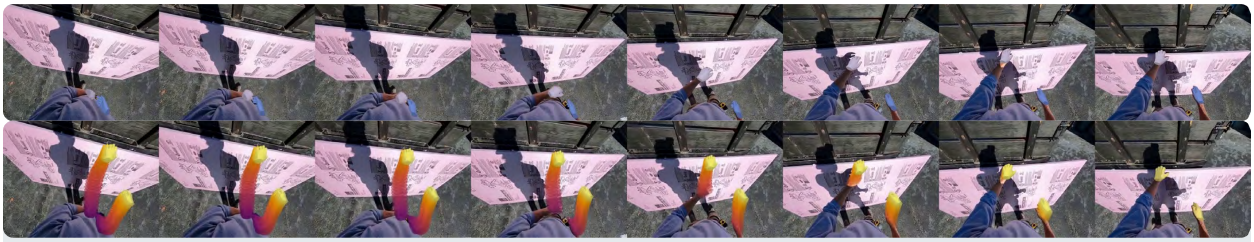
Left hand: *Slide the wooden stick across the table.*

Left hand: *Place the plate with food into the microwave.* ***Right hand:*** *Remove the pink plate from the microwave.*

Figure III: Examples of our hand V-L-A data used for pretraining. The first row in each case visualizes the reconstructed 3D hands for individual frames, while the second row shows the hand action trajectory from the current frame to the end of the episode. The color gradient from purple to yellow indicates the temporal progression from the beginning to the end of the episode.

Left hand: *Turn a page of the book with the left hand.*

Left hand: *Push the cabinet door open.*



Left hand: *Press the left hand against the top edge of the pink foam board.* ***Right hand:*** *Grasp the top edge of the pink foam board.*

Right hand: *Pick up the spray bottle.*

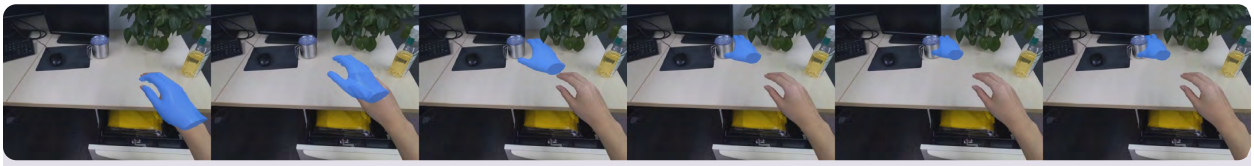
Left hand: *Pull and extend the tape measure along the edge of the wooden plank.* ***Right hand:*** *Place the tape measure onto the wooden plank.*

Figure IV: Examples of our hand V-L-A data used for pretraining. The first row in each case visualizes the reconstructed 3D hands for individual frames, while the second row shows the hand action trajectory from the current frame to the end of the episode. The color gradient from purple to yellow indicates the temporal progression from the beginning to the end of the episode.

Right hand: *Pick up the water filter.*

Right hand: *Pick up the green sauce bottle.*

Right hand: *Pick up the plate in the microwave oven.*



Right hand: *Pick up the silver cup.*

Right hand: *Grasp the cardboard box.*

Right hand: *Grasp the headphones.*

Left hand: *Pick up the yellow scissors.*

Left hand: *Pick up the metal basin.*

Figure V: Hand action prediction (grapsing) in unseen real-world environments by our method, with time increasing from left to right.

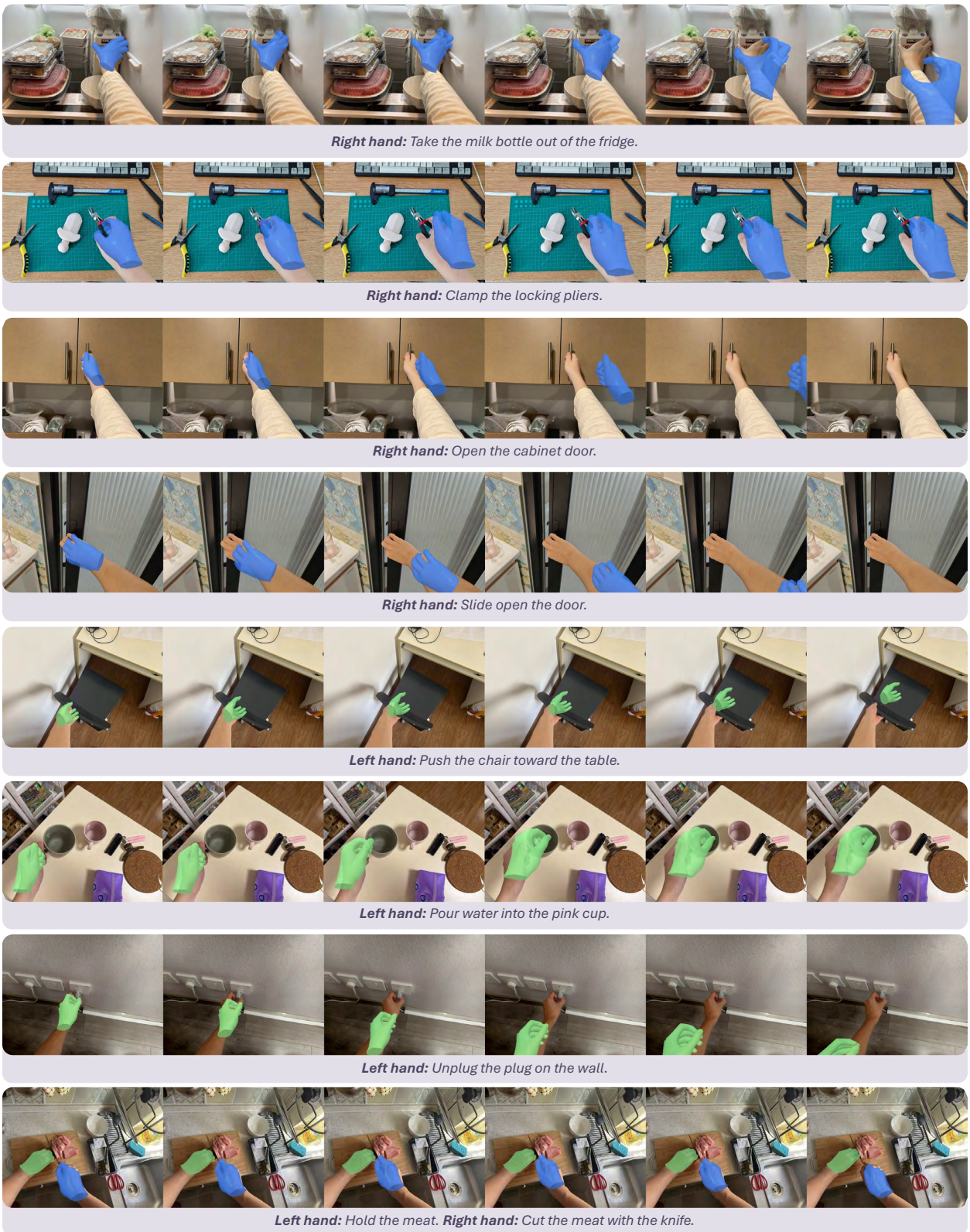


Figure VI: Hand action prediction (general action) in unseen real-world environments by our method, with time increasing from left to right.

Figure VII: In-domain execution trajectories of the *general pick-and-place* task and the *functional grasping* task. Rows 1–5 show examples of execution trajectories for the *general pick-and-place* task, while rows 6–8 present examples for the *functional grasping* task. Images are captured with the robot head camera, with time increasing from left to right.

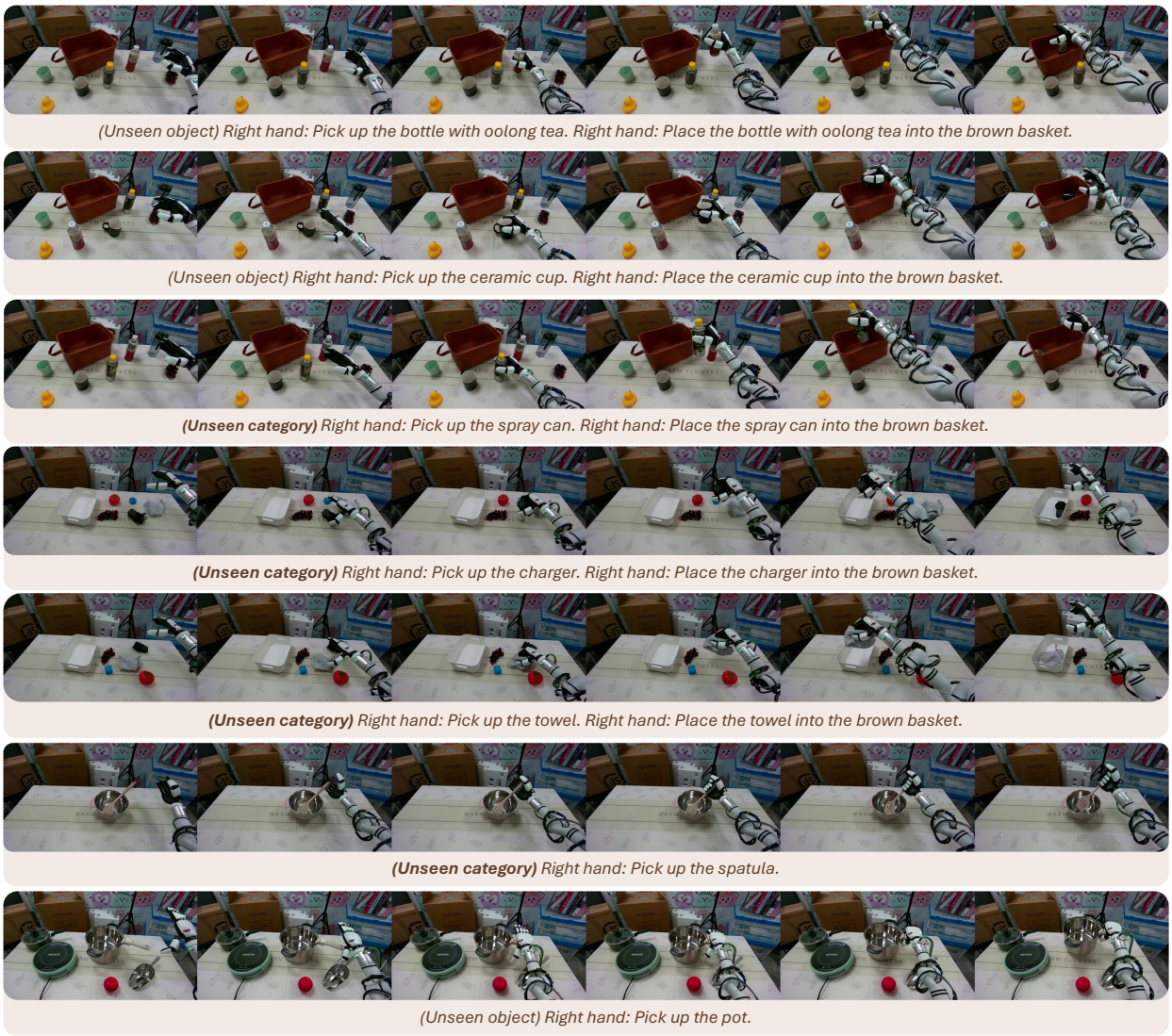
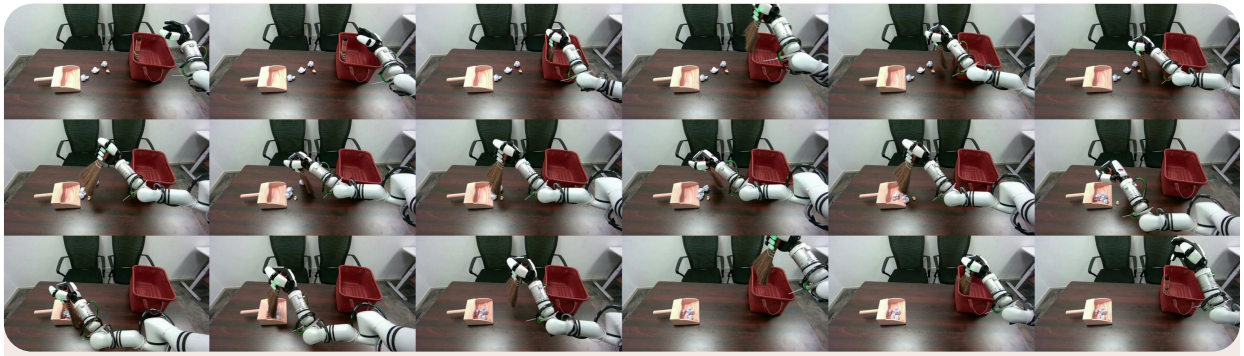


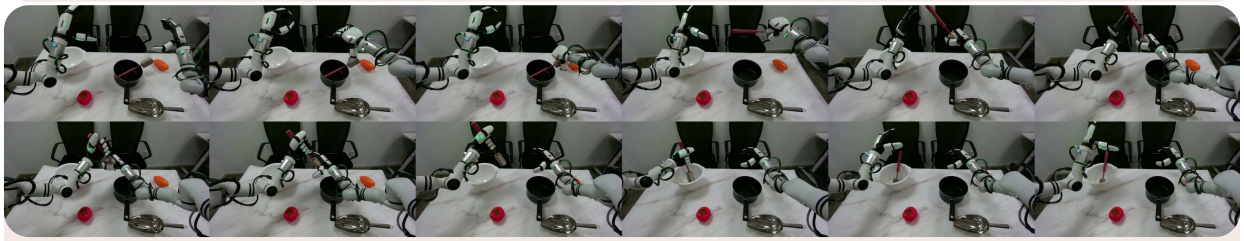
Figure VIII: Execution trajectories of the *general pick-and-place* task and the *functional grasping* task with *unseen background and objects*. Rows 1–5 show examples of execution trajectories for the *general pick-and-place* task, while rows 6–7 present examples for the *functional grasping* task. *Unseen object* means the objects are new but other objects of the same categories were seen in fine-tuning; and *unseen categories* means the objects belong to categories not encountered before. Images are captured with the robot head camera, with time increasing from left to right.



1. Right hand: Pick up the broom by its handle. 2. Right hand: Sweep the paper ball into the dustpan. 3. Right hand: Place the broom in the basket.

1. Right hand: Pick up the bottle. 2. Right hand: Pour the contents of the bottle into the pot. 3. Right hand: Place the bottle onto the table.

(Unseen object) 1. Right hand: Pick up the bottle. 2. Right hand: Pour the contents of the bottle into the pot. 3. Right hand: Place the bottle onto the table.



1. Right hand: Pick up the cup brush by the sponge. 2. Left hand: Receive the cup brush by the handle;
Right hand: Hand over the cup brush from the right hand to the left hand. 3. Left hand: Place the cup brush into the mixing bowl.

Figure IX: Execution trajectories of the sequential tasks. Rows 1–3 correspond to the *sweeping* task, rows 4–5 correspond to the *seen pouring* task, rows 6–7 correspond to the *pouring* task with *unseen backgrounds and objects*, and rows 8–9 correspond to the *bimanual dexterous handover* task. The numbers indicate the execution order of these tasks. Images are captured with the robot head camera, with time increasing from left to right and from top to bottom.