# VISION-LANGUAGE-ACTION INSTRUCTION TUNING: FROM UNDERSTANDING TO MANIPULATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

To operate effectively in the real world, robots should integrate multimodal reasoning with precise action generation. However, existing vision-language-action (VLA) models often sacrifice one for the other, narrow their abilities to task-specific manipulation data, and suffer catastrophic forgetting of pre-trained vision-language capabilities. To bridge this gap, we introduce **InstructVLA**, an end-to-end VLA model that preserves the flexible reasoning of large vision-language models (VLMs) while delivering leading manipulation performance with the help of embodied reasoning. InstructVLA introduces a novel training paradigm, *Vision-Language-Action Instruction Tuning (VLA-IT)*, which employs multimodal training with mixture-of-experts adaptation to jointly optimize embodied reasoning and action generation on both standard VLM corpora and a curated 650K-sample VLA-IT dataset. On in-domain SimplerEnv tasks, InstructVLA achieves 33.3% improvement over SpatialVLA. To evaluate generalization, we introduce SimplerEnv-Instruct, an 80-task benchmark requiring closed-loop control and high-level instruction understanding, where it outperforms a fine-tuned OpenVLA by 96% and an action expert aided by GPT-4o by 29%. Additionally, InstructVLA surpasses baseline VLMs on multimodal tasks and exhibits inference-time scaling by leveraging textual reasoning to boost manipulation performance in both simulated and real-world settings. These results demonstrate InstructVLA's potential for bridging intuitive and steerable human-robot interaction with efficient policy learning.

## 1 INTRODUCTION

Large-scale pretraining has produced versatile foundation models in computer vision (CV) (Oquab et al., 2023; Radford et al., 2021) and natural language processing (NLP) (Bai et al., 2023; Touvron et al., 2023). Building on this progress, recent Vision-Language-Action (VLA) models (Black et al., 2024; Kim et al., 2024) adapt large vision-language models (VLMs) (Karamcheti et al., 2024; Beyer et al., 2024) and finetune them on embodied datasets to achieve generalizable manipulation. While the integration of multimodal reasoning has led to significant advances in VLMs (Wei et al., 2022; Liu et al., 2024a), such reasoning remains largely unexplored in VLA settings. Fully leveraging VLMs for reasoning-guided manipulation beyond VLA initialization remains an open challenge. Current attempts to incorporate the reasoning capabilities of VLMs into action learning face three main obstacles: (1) task interference, catastrophic forgetting (French, 1999) of multimodal ability during action training; (2) data scarcity, particularly the limited availability of manipulation datasets with rich multimodal supervision; and (3) methodological gaps, specifically the lack of effective mechanisms and training paradigm to translate multimodal reasoning into action generation.

To address these challenges and utilize VLMs more effectively, prior work has primarily adopted two strategies. The first aims to retain general multimodal capabilities while learning manipulation skills through unified auto-regressive modeling. Models such as RT-2 (Brohan et al., 2023) and Magma (Yang et al., 2025) follow this approach by co-training on vision-language and manipulation data. Yet, this paradigm often overlooks complex embodied reasoning, and our ablations reveal that the general VLM corpus exhibits a domain gap in embodied scenarios. The second strategy tightly integrates embodied reasoning into manipulation datasets to transfer VLM capabilities. Methods such as ECoT (Zawalski et al., 2024) and Emma-X (Sun et al., 2024) embed chain-of-thought (CoT) reasoning into manipulation datasets. While promising, these methods rely on action-pretrained architectures (Kim et al., 2024) and structured reasoning formats (e.g., plans, subtasks, grounding),

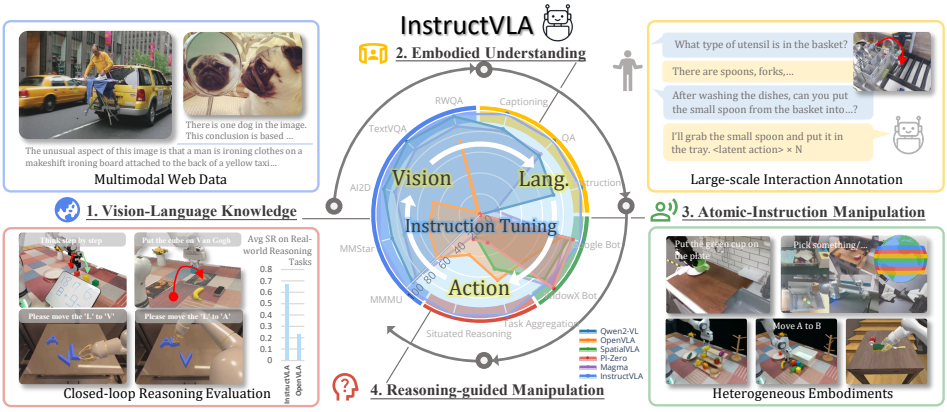



Figure 1: **Method overview.** InstructVLA integrates vision-language understanding with precise robotic control to achieve reasoning-guided manipulation. Its core training strategy, vision-language-action instruction tuning, enhances manipulation by unifying general multimodal knowledge, embodied reasoning, and atomic instruction-based manipulation into a coherent chain of thought.

which limit expressiveness, suffer from catastrophic forgetting, and fail to recover general multimodal capabilities-even with additional finetuning. Consequently, the extent to which VLM capabilities can be effectively translated into action generation in embodied contexts remains largely unexplored.

Building on these observations, we propose **InstructVLA**, a generalist VLA model that extends pretrained VLMs for accurate action generation while preserving strong multimodal understanding. Building on this unified modeling, we conduct extensive experiments to investigate how multimodal capabilities contribute to manipulation. Motivated by these insights, we design a training paradigm specifically tailored to bridge vision-language knowledge with action generation, treating language-conditioned action generation as an integral component of instruction following, as illustrated in Figure 1. To support this paradigm, we curate the **Vision-Language-Action Instruction Tuning (VLA-IT) dataset**, consisting of 650K human-robot interactions annotated with diverse instructions, scene captions, and question-answer pairs grounded in high-quality manipulation tasks (Ebert et al., 2021; Brohan et al., 2022). The training process follows a two-stage paradigm: (1) *Action Pretraining*, which trains a VLM-driven action expert using latent action representations distilled from language-based motion descriptions, while preserving the VLM backbone's multimodal capabilities; and (2) *Vision-Language-Action Instruction Tuning*, which unifies language and latent action generation through a trainable mixture-of-experts(MoE) adaptation framework. This framework is jointly trained on multimodal datasets (He et al., 2024), manipulation datasets, and the curated VLA-IT corpus, enabling the automatic switch between textual reasoning and action generation, thereby effectively leveraging vision-language understanding and reasoning for action generation.

To validate the performance of InstructVLA, we introduce the **SimplerEnv-Instruct benchmark**, a manually designed evaluation suite featuring 80 zero-shot manipulation tasks. It encompasses both closed-loop manipulation tasks and high-level instruction reasoning, involving either situated understanding or decomposition into actionable subtasks. With its thinking ability during manipulation, InstructVLA outperforms the fine-tuned OpenVLA baseline by 96% and achieves a 29% improvement over an action expert model assisted by GPT-4o on SimplerEnv-Instruct, demonstrating its effectiveness in instruction following and task decomposition. Furthermore, InstructVLA surpasses similarly sized VLMs in multimodal performance and shows a 33.3% improvement over SpatialVLA in closed-loop manipulation (Li et al., 2024d). Our contributions can be summarized as follows:

- **Model.** We propose **InstructVLA**, a VLA architecture and training pipeline that **supports studying language capability in VLAs** by efficiently preserving pretrained vision-language knowledge from VLMs while integrating manipulation as a component of instruction following.
- **Dataset & Benchmark.** We design a **practical data and evaluation pipeline** for vision-language-action instruction following, supported by 650K tailored VLA-IT annotations and a manually curated benchmark suite, enabling evaluation of VLAs' instruction generalization capabilities.
- **Validation.** InstructVLA achieves leading performance across robotic manipulation tasks, multimodal benchmarks, and real-world deployments, enabling intuitive and controllable manipulation.

## 2 Related Works

**Policy learning at scale.** Following the success of CV (Oquab et al., 2023; Zhai et al., 2023)and NLP (Touvron et al., 2023), recent research (Wang et al., 2024a; Brohan et al., 2022; 2023; Zheng et al., 2025; Wang et al., 2024b; Niu et al., 2025) shows that robot policies improve when trained in large heterogeneous datasets. RT-1 (Brohan et al., 2022) and RT-2 (Brohan et al., 2023), trained in large-scale real-world demonstrations, achieve strong in-domain accuracy and zero-shot transfer. Works such as Octo (Octo Model Team et al., 2024) and RT-X (Collaboration et al., 2023) extend this approach by aggregating the largest open-source manipulation datasets (Collaboration et al., 2023). Some methods, such as LAPA (Ye et al., 2024), Seer (Tian et al., 2024), and Moto (Chen et al., 2024b), use video generation and inverse dynamics to learn scalable motor representations. In the VLA domain, models are typically initialized from pretrained vision-language models (Kim et al., 2024; Qu et al., 2025; Brohan et al., 2023) leveraging prior visual-linguistic alignment instead of learning from scratch. Further, methods such as RT-Trajectory (Gu et al., 2023) and GraspVLA (Deng et al., 2025b) jointly train intermediate manipulation representations such as trajectories or bounding boxes using a combination of real and simulated data to guide action generation and enhance generalization.

**Vision-language-action models.** Recent foundation models (Brohan et al., 2023; Kim et al., 2024; Qu et al., 2025; Black et al., 2024; Chen et al., 2024b; Bjorck et al., 2025; Pertsch et al., 2025; Niu et al., 2024) integrate perception, language, and robot manipulation into a single network, using two main architectures. Autoregressive models such as RT-2 (Brohan et al., 2023), OpenVLA (Kim et al., 2024) and SpatialVLA (Qu et al., 2025) treat actions as discrete tokens. LLARVA (Niu et al., 2024) introduces 2D trace for pretraining. FAST tokenization (Pertsch et al., 2025) further compresses motion sequences. In contrast, flow-based VLAs avoid discretization; for example, $\pi_0$ (Black et al., 2024) and GR00T (Bjorck et al., 2025) generate actions through continuous flow matching (Lipman et al., 2022), while CogACT (Li et al., 2024a) and CronusVLA (Li et al., 2025a) use diffusion (Peebles & Xie, 2023). Hybrid approaches, like RoboDual (Bu et al., 2024), combine generalist action models with specialist action experts. Although flow-based methods (Black et al., 2024; Bjorck et al., 2025; Li et al., 2025a; 2024a) often achieve superior performance, they typically neglect the integration of autoregressive text reasoning (Brohan et al., 2023), which is crucial for leveraging the VLM's semantic capabilities. In contrast, our model unifies autoregressive VLM language generation with the flow-based action generation, demonstrating efficient co-training of language and action.

**Bringing reasoning ability to manipulation.** Bridging pre-trained world knowledge to enhance the generalization of robot policies is a promising direction. One line of work standardizes intermediate representations, such as primitive (Chen et al., 2024c), trajectories (Gu et al., 2023; Li et al., 2025b), keypoints (Li et al., 2024b) and masks (Huang et al., 2025). However, these approaches often rely on rule-based decomposition and hand-crafted planning heuristics, whose rigid separation from low-level control limits scalability and hinders end-to-end policy learning. More recently, unified modeling of perception, reasoning, and manipulation (Brohan et al., 2023; Intelligence et al.; AI, 2024; Shentu et al., 2024; Belkhale et al., 2024), along with other generative formulations (Pan et al., 2025; Zhou et al., 2024), has demonstrated the potential of leveraging pre-trained VLMs and LLMs for reasoning-guided generation, revealing emerging capabilities (Deng et al., 2025a). Yet, many prior studies depend on closed-source data (Intelligence et al.) or restrict evaluation to real-world settings (Brohan et al., 2023; Belkhale et al., 2024; Zhou et al., 2025), limiting reproducibility and large-scale assessment. Our work provides an initial exploration, supported by open data and benchmarks, to study *reasoning-guided manipulation* through the integration of reasoning and action.

## 3 InstructVLA

We propose **InstructVLA** (Figure 2), a unified model for joint language-action generation that also mitigates task interference and catastrophic forgetting. Section 3.1 describes the architecture, including dynamic switching between reasoning and execution modes, as well as inference strategies. Section 3.2 presents the training paradigm for VLA instruction following.

### 3.1 Architecture

**Embodied VLM for textual and latent action generation.** We propose a unified framework that enables simultaneous multimodal reasoning and language-steered latent action planning using a single

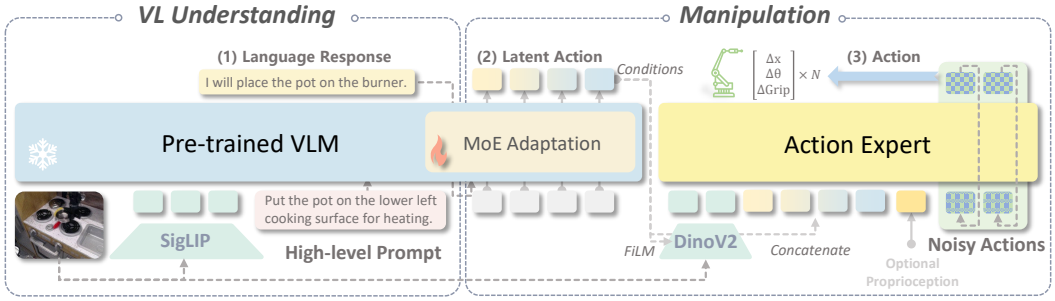



Figure 2: **Overview of the InstructVLA.** InstructVLA integrates the multimodal reasoning capabilities of a vision-language model with robotic manipulation. Generation consists of three steps: (1) asynchronous auto-regressive reasoning by the VLM, (2) latent action generation, and (3) action decoding. A MoE adaptation enables the VLM to alternate between reasoning and latent action prediction. The flow matching action expert decodes the final actions, conditioned on latent actions.

VLM (Figure 2 (1) and (2)). The model produces textual outputs to preserve the strong language understanding and multimodal inference capabilities of the pretrained VLM, while subsequently generating latent action representations for downstream manipulation. To support action planning, we introduce $N$ learnable action queries $Q \in \mathbb{R}^{N\times D}$, which attend to the VLM's hidden states and extract task-relevant latent action $C \in \mathbb{R}^{N\times D}$, where $D$ is the VLM hidden dimension. Our implementation builds on the compact and efficient Eagle2-2B backbone (Li et al., 2025c), with a tailored training strategy described in Section 3.2. The VLM is supervised with cross-entropy on language output with loss $\mathcal{L}_{LM}$.

**MoE adaptation to harmonize reasoning and action.** A key challenge is enabling the model to seamlessly alternate between reasoning and manipulation. To this end, we adopt a Mixture-of-Experts (MoE) design (Zhou et al., 2022), which allows adaptive reweighting of expert modules based on input context and reasoning mode, thereby integrating multimodal reasoning with language-steered latent action. Specifically, LoRA (Hu et al., 2022) modules are employed as experts within the LLM backbone, preserving pretrained capabilities while ensuring efficient inference. A scale head (E.L. Buehler, 2024) predicts gating coefficients $\lambda_i$ for each expert by classifying the hidden state, enabling the model to adaptively blend their outputs. The resulting hidden states for $K$ experts are computed as $h = W_0 x + \sum_{i=0}^{K} B_i A_i x \cdot \alpha_i \cdot \lambda_i$, where $W_0$ is the original weight, $x$ denotes input, $A_i \in \mathbb{R}^{r\times d}$ and $B_i \in \mathbb{R}^{d\times r}$ are the LoRA parameters, $\alpha_i$ is the LoRA scaling factor.

**Flow model as an efficient action expert.** To further decouple low-level control from high-level understanding, the action expert is designed to generate actions from image observations conditioned on VLM-derived intentions. It takes image features from DINOv2 (Oquab et al., 2023) vision encoder, latent actions, noisy action embeddings and optional information such as proprioception, and fuses these with a simple transformer architecture (Touvron et al., 2023) with block-wise causal attention. Specifically, non-causal attention is applied within each input, and causal attention between input types. The vision encoder, further enhanced with feature-wise linear modulation (FiLM) (Perez et al., 2018), plays a crucial role in directing actions to spatial and contextual input. The flow matching objective (Black et al., 2024) is used to supervise action learning, as detailed in Section F.2.

**Inference.** InstructVLA integrates language and action generation in a single model with the following techniques to improve speed. (1) *Decoding strategies.* To mitigate the latency of autoregressive decoding, textual responses are generated via greedy search until the first action query token appears. The remaining action queries are then decoded in parallel within a single forward pass of the VLM. (2) *Language response and latent action caching.* We decouple language response from action generation by caching textual outputs across multiple action steps, leveraging their temporal stability. InstructVLA also supports cache latent actions, which reduces the number of VLM forward with minimal performance impact compared with ECoT (Zawalski et al., 2024) (see Section A.1).

### 3.2 Training Recipe

Direct co-training of vision, language, and action often leads to unstable optimization and slow convergence. We therefore adopt a principled two-stage training paradigm: first, action pretraining to

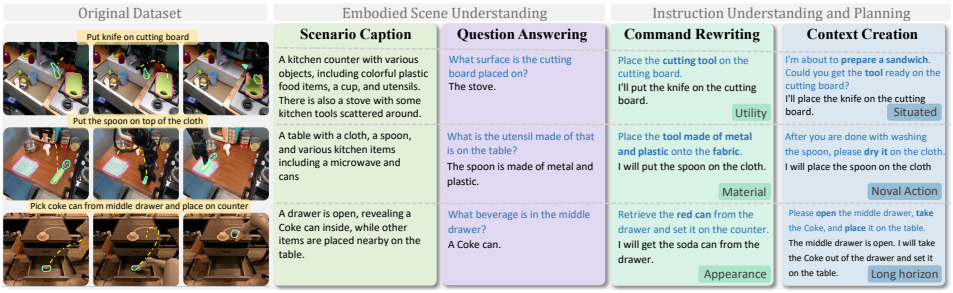



Figure 3: **Vision-language-action instruction tuning data examples.** Annotations focus on: (1) improving scene understanding and (2) learning instruction following and planning.

align with the VLM's latent action embeddings; second, vision-language-action instruction tuning to integrate multimodal reasoning with manipulation.

**Stage 1: Action pre-training.** InstructVLA is pre-trained using heterogeneous manipulation data (Brohan et al., 2022; Ebert et al., 2021). To distill the knowledge from the VLM for manipulation, the model is trained to predict both actions and language motion (Section 4.1), with the latter supervised via cross-entropy loss. Due to the stability of flow matching and the next token prediction, the final loss is the direct sum of both losses as $\mathcal{L} = \mathcal{L}_{LM} + \mathcal{L}_{FM}$. During this stage, only the input and output embedding of the action queries and action LoRA adapter on the LLM backbone are tuned, consisting of 650M parameters. The model trained is named the "Expert".

**Stage 2: Vision-language-action instruction tuning.** We extend the concept of visual instruction tuning (Liu et al., 2023) with a simple approach to train InstructVLA. Our observation is that once the action expert is pretrained to follow the latent actions from the VLM, further adapting the LLM backbone enables the model to handle manipulation tasks with more complex instructions and generate appropriate responses. In this stage, the action expert remains frozen; a new language LoRA adapter and scale head of the MoE-adaptation are added. The MoE module is the only trainable parts, comprising 220M parameters. We detail the data pipeline for vision-language-action instruction tuning in Section 4.1; this data bridges pretrained vision-language capabilities with embodied task scenarios. We further co-train the model using multimodal datasets (He et al., 2024) to bootstrap multimodal understanding. The resulting model is referred to as the "Generalist", reflecting its combined vision-language and manipulation capabilities.

# 4 VLA Dataset and Benchmark

## 4.1 InstructVLA Tuning Dataset

We curate diverse hierarchical language annotations from large-scale manipulation datasets (Brohan et al., 2022; Ebert et al., 2021), including language motion (Belkhale et al., 2024) as detailed in Section D.1, along with the VLA-IT dataset for instruction tuning and reasoning transferring.

**Vision-language-action instruction tuning data.** To enable language-steerable VLA models, it is essential to curate diverse instructions, model responses, and reasoning patterns. We categorize our data into four types as illustrated in Figure 3. For embodied scene understanding: *(1) Scenario captioning* provides descriptions of the robot's environment *(2) Question answering* targets scene understanding through consistent QA pairs across an episode. Together, they bridge vision-language annotations with embodied scenes. For instruction understanding and latent action planning: *(3) Command rewriting* introduces instructional diversity through paraphrasing, attribute-based references and varied vocabulary. *(4) Context creation* generates implicit user goals or progress cues in multi-step tasks, requiring the robot to infer intent. These annotations support joint VLA reasoning.

We use GPT-4o (OpenAI, 2023) to annotate data with three frames from each episode, along with the corresponding instruction. Ground-truth instruction is crucial for annotation accuracy, emphasizing that even state-of-the-art VLMs can make errors in embodied tasks, leading to a performance gap when using GPT-4o as an instruction interpreter for such tasks. Additional details of the dataset analysis and prompt templates are provided in Section D.

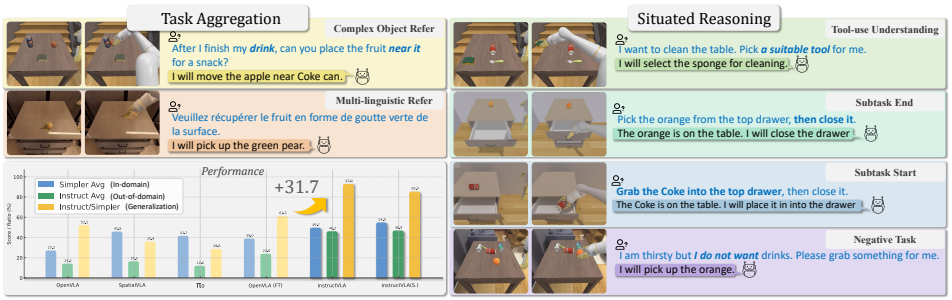



Figure 4: **Simpler-*Instruct*.** Six representative test cases with instructions and InstructVLA responses. Prior VLAs exhibit limited generalization compared to InstructVLA.

### 4.2 SimplerEnv-Instruct

Building upon the SimplerEnv platform, we introduce **SimplerEnv-Instruct**, a benchmark specifically designed to evaluate the instruction-following and reasoning capabilities of vision-language-action (VLA) models in a zero-shot setting. Unlike prior manipulation benchmarks that primarily focus on atomic actions or low-level control, SimplerEnv-Instruct captures two essential yet underexplored abilities: (1) policy generalization to linguistic and visual diversity, and (2) contextual reasoning in situated environments, evaluated in the *situated reasoning* suite.

**Task creation.** We remove trivial cases and design novel tasks requiring genuine generalization rather than memorization. Novel objects and instructions are strictly out-of-distribution from the originals, and all tasks are cross-validated by three annotators for clarity and consistency. In total, we curated 80 tasks with 1.1K trials, about one third the size of SimplerEnv, keeping evaluation practical.

- **Task aggregation.** (50 tasks; examples shown in Figure 4, left). This suite assesses a model's ability to consistently interpret and execute core tasks based on both instructions and environmental context, despite variations in visual or linguistic forms. Tasks cover phenomena such as novel verbs, multilingual expressions, diverse object references, sentence rephrasings, and OOD objects.
- **Situated reasoning.** (30 tasks; examples shown in Figure 4, right). Beyond *task aggregation*, this suite evaluates a model's ability to reason over contextual cues or indirect instructions and to decompose commands into sub-goals. For example, "I want to clean the table. Pick a suitable tool for me." requires selecting the correct object (e.g., a sponge) from context.

Together, by leveraging the large-scale real-world training dataset, **SimplerEnv-Instruct** provides a reproducible benchmark that evaluates VLA generalization to unseen tasks. It achieves an affordable evaluation cost while systematically probing both task generalization and reasoning, filling a critical gap in VLA evaluation with a diagnostic, human-interpretable, and standardized benchmark.

## 5 Experiment

**Benchmarks.** *(a) Multimodal:* We adopt automatic evaluation from VLMEvalKit (Duan et al., 2024), as detailed in Section E.1. *(b) SimplerEnv:* This benchmark (Li et al., 2024d) provides real-to-sim evaluation on large-scale manipulation datasets, incorporating visual matching and variance aggregation to assess generalization. *(c) SimplerEnv-Instruct:* As described in Section 4.2, this extension of SimplerEnv introduces novel objects, tasks, and instructions, offering a broader testbed for evaluating instruction generalization in VLAs. In addition, we assess embodied understanding in Section A.2 and manipulation performance on the LIBERO (Liu et al., 2024b) benchmark in Section A.3.

**Training details.** The VLM is trained with a resolution of $448 \times 448$ following Li et al. (2025c), while the action expert operates at $224 \times 224$ as in (Kim et al., 2024), using a fixed learning rate of 5e-5 without warm-up. The action expert employs a 12-layer transformer backbone with a hidden size of 768. Following Black et al. (2024), a $\beta$ distribution is used to enhance accuracy on the noisier time steps. During Stage 2 finetuning, manipulation and multimodal understanding are trained in an interleaved manner. Owing to InstructVLA's training paradigm, multimodal capabilities are preserved easily. We adopt a 1:7 ratio, twice the imbalance of ECoT and ChatVLA (1:3), reducing the additional computation needed to maintain multimodal ability. Further details are provided in Section F.

Table 1: **Multimodal understanding.** #Params is the size of LLM backbone. S. denotes robot state.

| Methods | #Params | Multi-modal Understanding Benchmarks | | | | | | VQA Benchmarks | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MMMU$^{Val}$ | MM-Vet | MMStar | MME$^{P}$ | OCRBench | HallB | MMB | TextVQA | DocVQA | InfoVQA | AI2D | ChartQA | RWQA |
| Bunny (He et al., 2024) | 8B | 43.4 | 39.1 | 45.4 | **1987.7** | 444 | 37.7 | 72.9 | - | - | - | 69.4 | 30.1 | 60.4 |
| PaliGemma (Beyer et al., 2024) | 2B | 34.9 | 33.1 | 48.3 | 1686.1 | 614 | 32.2 | 65.6 | 68.1 | 74.0 | 34.0 | 68.3 | 33.1 | 55.2 |
| Eagle2 (Li et al., 2025c) | 1.5B | 43.1 | **53.8** | **56.4** | 1572.1 | <u>818</u> | **45.8** | 74.9 | 79.1 | <u>88.0</u> | 65.8 | **79.3** | <u>82.3</u> | <u>63.1</u> |
| Qwen2-VL (Wang et al., 2024c) | 1.5B | 41.1 | 51.5 | 48.0 | <u>1872.0</u> | 809 | 41.7 | 74.9 | 74.9 | **88.6** | 61.4 | 74.7 | 73.5 | 62.9 |
| OpenVLA (Kim et al., 2024) | 7B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OpenVLA (FT) | 7B | 26.0 | 9.1 | 28.2 | 87.6 | 2.5 | 8.4 | 18.9 | 2.5 | 29.2 | 43.4 | 35.8 | 1.4 | 47.2 |
| ECoT (Zawalski et al., 2024) | 7B | 16.2 | 0.0 | 19.1 | 0.0 | 0.0 | 3.1 | 0.9 | 0.0 | 2.2 | 0.0 | 0.0 | 0.0 | 29.8 |
| ChatVLA Zhou et al. (2025) | 1.5B | 37.4 | - | 47.2 | 1435.2 | 729 | 39.9 | 69.0 | 71.2 | 83.3 | 53.3 | 67.6 | 59.9 | 57.0 |
| Magma (Yang et al., 2025) | 8B | 38.8 | 34.1 | 41.3 | 1496.5 | 518 | 38.0 | 69.7 | 66.5 | 65.4 | 45.2 | 66.1 | 61.8 | 56.5 |
| **InstructVLA-Generalist** | 1.5B | **44.2** | 51.7 | <u>56.2</u> | 1529.6 | 814 | <u>45.6</u> | <u>76.1</u> | <u>77.7</u> | 85.8 | **63.7** | <u>79.1</u> | 81.7 | <u>63.1</u> |
| **InstructVLA-Generalist(S.)** | 1.5B | <u>43.8</u> | **54.0** | 56.0 | 1548.0 | **829** | 42.8 | **76.3** | **78.2** | 86.0 | **63.7** | 78.9 | **82.9** | **63.5** |

Table 2: **Robotic manipulation.** Google and WidowX Robot denote two embodiments in SimplerEnv. For SimplerEnv-Instruct, we focus on two reasoning levels instead of embodiments. Magma$^{\dagger}$ denotes evaluation with sampling. The results of InstructVLA are averaged over three random seeds.

| Methods | Google Robot | | | | | | | | WidowX Robot | | | Avg | SimplerEnv-Instruct | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Open/Close Drawer | | Put in Drawer | | Pick Coke Can | | Move Near | | Put Spoon | Put Carrot | Stack Blocks | | Task Aggregation | Situated Reasoning | Avg |
| | VM | VA | VM | VA | VM | VA | VM | VA | | VM | | | | | |
| RT-1-X (Collaboration et al., 2023) | 59.7 | 29.4 | 21.3 | 10.1 | 56.7 | 49.0 | 31.7 | 32.3 | 0.0 | 4.2 | 0.0 | 26.8 | - | - | - |
| RT-2-X (Collaboration et al., 2023) | 25.0 | 35.5 | 3.7 | 20.6 | 78.7 | 82.3 | 77.9 | 79.2 | - | - | - | - | - | - | - |
| RoboVLMs-2B (Li et al., 2024c) | 43.5 | 10.6 | 27.8 | 0.0 | 77.3 | 75.6 | 61.7 | 60.0 | 45.8 | 20.8 | 4.2 | 38.8 | - | - | - |
| OpenVLA-7B (Kim et al., 2024) | 63.0 | 28.8 | 0.0 | 0.0 | 18.0 | 60.8 | 56.3 | 67.7 | 4.2 | 0.0 | 0.0 | 27.2 | 14.8 | 13.6 | 14.2 |
| SpatialVLA-3B (Qu et al., 2025) | 57.4 | 41.8 | 0.9 | 9.1 | 86.0 | 88.0 | 77.9 | 72.7 | 16.7 | 25.0 | 29.2 | 45.9 | 23.6 | 9.8 | 16.5 |
| GR00T-N1.5-3B (Bjorck et al., 2025) | 27.8 | 13.2 | 7.4 | 2.2 | 51.7 | 63.6 | 51.0 | 54.0 | 62.5 | 45.8 | 16.7 | 36.0 | - | - | - |
| $\pi_0$-3B (Black et al., 2024) | 64.8 | 48.4 | 13.9 | 15.4 | 70.3 | 44.7 | 41.0 | 35.5 | 37.5 | 50.0 | 37.5 | 41.7 | 12.1 | 11.8 | 12.0 |
| **InstructVLA-Expert** | 52.3 | 61.7 | 50.3 | 33.1 | 79.6 | 92.3 | 68.3 | 71.9 | 43.1 | 40.4 | 9.7 | <u>50.9</u> | $21.6 \pm 1.4$ | $12.9 \pm 0.4$ | <u>17.3</u> |
| **InstructVLA-Expert(S.)** | 46.8 | 54.1 | 45.7 | 70.0 | 96.0 | 95.9 | 79.7 | 82.4 | 61.1 | 54.2 | 36.1 | **61.2** | $20.9 \pm 0.3$ | $20.5 \pm 1.0$ | **20.7** |
| Magma-8B (Yang et al., 2025) | 9.7 | 5.8 | 0.0 | 0.0 | 46.0 | 46.4 | 60.0 | 82.0 | 45.8 | 33.3 | 8.3 | 30.5 | 15.5 | 9.9 | 12.7 |
| Magma-8B$^{\dagger}$ (Yang et al., 2025) | 56.0 | 53.4 | 6.4 | 18.5 | 83.7 | 68.8 | 65.4 | 65.7 | 35.5 | 31.0 | 12.7 | 43.6 | 26.2 | 21.4 | 23.8 |
| OpenVLA (FT) 7B | 63.9 | 42.6 | 3.7 | 6.9 | 62.3 | 88.7 | 65.8 | 67.7 | 12.5 | 33.3 | 4.2 | 39.0 | 28.3 | 19.5 | 23.9 |
| OpenVLA (FT&GPT) | - | - | - | - | - | - | - | - | - | - | - | - | 38.8 | 32.4 | 35.6 |
| **InstructVLA-Generalist** | 64.5 | 61.7 | 38.3 | 27.5 | 81.7 | 91.8 | 55.8 | 69.7 | 31.9 | 34.7 | 12.5 | <u>49.7</u> | $43.6 \pm 1.4$ | $48.8 \pm 0.8$ | <u>46.2</u> |
| **InstructVLA-Generalist(S.)** | 39.8 | 51.1 | 45.7 | 57.3 | 91.0 | 93.0 | 71.7 | 78.3 | 62.4 | 48.6 | 15.3 | **54.9** | $48.2 \pm 1.3$ | $45.6 \pm 0.5$ | **46.9** |

**Baselines.** We categorize the baselines into three groups: (1) *Multimodal VLMs*, including Bunny(He et al., 2024), PaliGemma (Beyer et al., 2024), Eagle2 (Li et al., 2025c), and Qwen2-VL (Wang et al., 2024c); (2) *VLA models*, including RT-1-X and RT-2-X (Collaboration et al., 2023), RoboVLMs (Li et al., 2024c), SpatialVLA (Qu et al., 2025), $\pi_0$ (Black et al., 2024), GR00T-N1.5 (Bjorck et al., 2025), and OpenVLA (Kim et al., 2024); (3) *Generalist VLA models*, including Magma (Yang et al., 2025), OpenVLA fine-tuned (FT) from generalist pretrained model on both robotic and multimodal data, and ECoT(Bridge) (Zawalski et al., 2024). During evaluation, InstructVLA and other baselines use a temperature of 0 without sampling to expedite generation. We re-evaluate Magma with official checkpoint[1]. For ECoT, we report only its multimodal results due to its real-to-sim domain gap.

### 5.1 Main Results

We present our main results in Tables 1 and 2. In Table 1, using the same generalist model InstructVLA (generalist), it not only outperforms the co-trained baseline Magma, but is also comparable to its base model Eagle2 and Bunny (VLM data corpus). InstructVLA further demonstrates stronger embodied understanding as detailed in Section A.2. In Table 2, InstructVLA (expert) outperforms the expert baseline SpatialVLA by 33.3% on SimplerEnv. Meanwhile, InstructVLA (generalist) not only maintains strong performance on SimplerEnv's atomic instructions but also achieves a 31.7% improvement on SimplerEnv-Instruct over the state-of-the-art baseline (OpenVLA with GPT-4o).

However, we observe that finetuning OpenVLA on multimodal and manipulation datasets does not fully restore its original multimodal capabilities, although it does improve task performance. Its performance can be further enhanced by integrating GPT-4o as an API-based system-2 module to rephrase instructions (OpenVLA (FT&GPT)). However, GPT-4o faces the same challenges in accurate instruction rewriting as noted in Section 4.1, and fails to outperform InstructVLA (Generalist). Methods such as Magma, which co-train both abilities of the VLM, better preserve multimodal ability, but still fail to match the performance of our approach. Although it also adapts two-stage training, ECoT relies solely on textual chain-of-thought reasoning over manipulation datasets and lacks the capability for multimodal question answering. We observe that it consistently generates manipulation-style CoT responses, without demonstrating effective instruction-following ability.

[1]We observe a notable performance gain for Magma when using sampling. Accordingly, we report its official score on SimplerEnv and re-evaluate its performance on SimplerEnv-Instruct under the sampling setting.

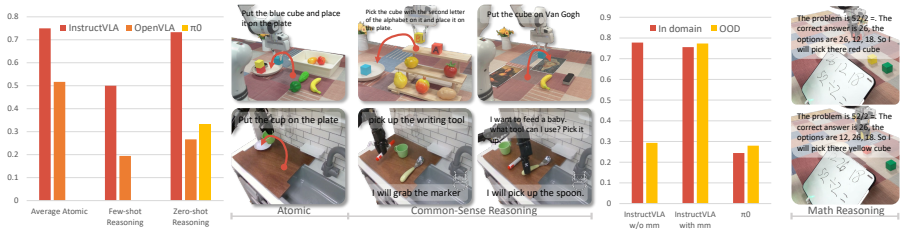



Figure 5: **Real-world experiments.** "Atomic" refers to atomic instructions. For the Kitchen and math settings, InstructVLA's responses are presented.

| Experts | WidowX Bot | Google Bot | Ave. |
|---|---|---|---|
| w/o DINO | 4.2 | 32.4 | 23.0 |
| w/o FiLM | 25.0 | 56.3 | 45.9 |
| w/o Lang. | 15.3 | 65.0 | <u>48.4</u> |
| InstructVLA | 29.1 | 64.8 | **52.9** |

Table 3: **Ablation of action expert vision design and language motion.** "w/o Lang." denotes without using language motion. "w/o FiLM" denotes using only DINO. "w/o DINO" denotes action expert without the vision input.

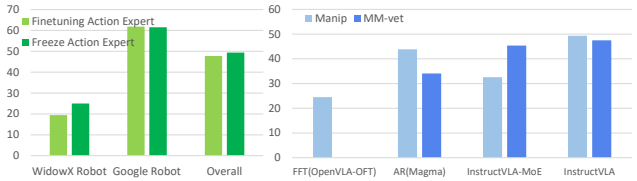



Figure 6: **Finetuning strategies.** (a) Freezing or finetuning the action head during VLA-IT training. (b) Training strategies when multimodal and manipulation tasks co-exist. "FFT" denotes full finetuning. "AR" denotes auto-regressive.

### 5.2 Real-world Experiments

To evaluate InstructVLA in real-world scenarios, we conduct zero-shot experiments on the WidowX-250 Arm and few-shot experiments on the Franka Research 3 robot, as shown in Figure 5. The few-shot tasks involve spatial pick-and-place from a rack and cluttered tabletop setting and math-centric tasks detailed in Section A.5 to demonstrate the role of multimodal data. The zero-shot tasks are set in a kitchen environment following the Bridge dataset. InstructVLA is fine-tuned using the proposed training recipe, while OpenVLA is jointly trained on atomic skill and VLA-IT datasets with extra language supervision. The $\pi_0$ is finetuned using the official repository.

Each scenario includes both atomic and reasoning instructions. Atomic tasks emphasize in-domain objects and instructions with a focus on spatial generalization to assess baseline VLA capabilities. Both models perform comparably on direct in-domain instructions, but InstructVLA achieves a 23.3% improvement over OpenVLA. For reasoning tasks such as celebrity recognition, OCR, and tool-use inference, OpenVLA shows a substantial performance drop, whereas InstructVLA outperforms it by 41.7% in few-shot and 46.7% in zero-shot settings. On reasoning and math tasks, InstructVLA achieves a 2.5× improvement over $\pi_0$, which behaves close to random guessing. Additional ablations and experimental setups are provided in Sections A.5 and H.

### 5.3 Ablation Studies

We conduct ablation studies guided by two central questions: (1) Section 5.3.1. How can manipulation and multimodal understanding be effectively integrated into a single model through architectural design and training strategies? (2) Section 5.3.2. To what extent does vision-language comprehension enhance manipulation performance in complex scenarios? Through targeted ablations, we examine the impact of key architectural and training decisions on these capabilities.

#### 5.3.1 Multimodal and Manipulation Co-training

**Strategies for multimodal and manipulation co-training.** As shown in Figure 6 (b), four paradigms are compared. (1) Following OpenVLA-OFT, FFT denotes full finetuning of the model with latent actions but without MoE adaptation and multi-stage training. With comparable computational resources, this setting yields suboptimal performance on both manipulation and understanding tasks. (2) The AR paradigm (Magma, RT-2) supports co-training but has limited performance. (3) Removing the MoE design while keeping the training paradigm preserves multimodal performance but reduces manipulation capability. (4) In contrast, InstructVLA leverages our proposed architecture and two-stage training strategy, achieving a 12.5% improvement over Magma on SimplerEnv.

| QA & Cap. | T.A. | S.R. | Ave. |
|---|---|---|---|
| ✗ | 40.7 | 42.7 | 41.7 |
| ✓ | 43.6 | 48.8 | 46.2 |

Table 4: **Effect of QA and caption data.** "T.A." denotes task aggregation. and "S.R." denotes situated reasoning on SimplerEnv-Instruct.

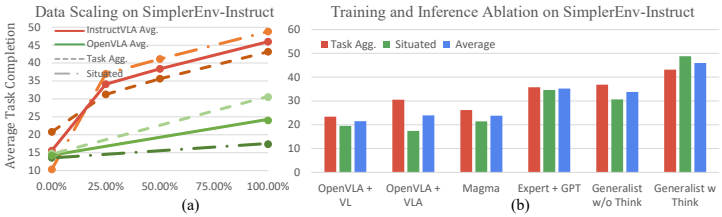



Figure 7: **Data scaling and multimodal training.** Impact of scaling and training strategies on manipulation with multimodal reasoning.

**Effects of language motion data for pre-training.** As shown in Table 3, introducing "language motion" (auxiliary textual descriptions of low-level actions) enhances the VLM's ability to associate visual cues with manipulation primitives, leading to a 9.3% improvement in overall success rate.

**Action expert perception design.** Incorporating richer perception into the action expert is efficient due to its compact design compared to the VLM backbone. As shown in Table 3, while the base VLM offers general visual understanding, fine-grained perception for manipulation tasks demands richer representations. Removing the DINOv2-based ViT encoder from the action expert results in a 50.0% performance drop, highlighting its critical role in capturing task-relevant visual cues. Incorporating FiLM to the ViT encoder yields a further 15.3% improvement by modulating visual features with latent actions. As shown in Table 2 the expert model with robot state generally performs better.

### 5.3.2 Multimodal Ability Transfers to Manipulation

**Ablation on VL-to-action learning.** As shown in Figure 6(a), we examine the effect of VLA instruction tuning by comparing two configurations: (1) finetuning only the VLM, and (2) jointly finetuning both the VLM and the action expert. Freezing the action expert achieves performance comparable to joint finetuning while substantially reducing the number of trainable parameters. This suggests that InstructVLA can effectively adapt to complex textual inputs by fine-tuning only the VLM, without altering the pretrained action expert.

**Effects of VLA-IT data.** As shown in Figure 7(a), we evaluate the **scaling behavior** of VLA-IT annotations on the SimplerEnv-Instruct benchmark. Situated reasoning tasks, which require grounding objects and goals in context, benefit most from larger annotation sets, highlighting the bootstrapped reasoning abilities inherited from VLMs. In contrast, pretrained OpenVLA fine-tuned on VLA-IT gains primarily from increased instruction diversity but shows limited improvement on situated reasoning tasks due to catastrophic forgetting of VL capabilities. These findings suggest that two-stage methods such as ECoT may be insufficient for fully leveraging the multimodal capacity of VLMs. We also examine the **effect of annotation diversity**, as shown in Table 4, where adding QA and captioning improves generalization by 10.8%. Additional ablations are provided in Section A.4.

**Training and inference strategies for reasoning-guided manipulation.** As shown in Figure 7 (b), (1) Simply combining manipulation and general multimodal ability through co-training does not yield significant benefits. Magma, despite co-training on multimodal datasets, shows limited transfer of vision-language capability to reasoning tasks on SimplerEnv-Instruct. Although OpenVLA suffers from catastrophic forgetting when finetuned with VLA-IT corpus, it still achieves better performance than Magma. (2) Multimodal ability can implicitly benefit manipulation when preserved through embodied reasoning annotation. Our generalist model, trained on the VLA-IT corpus, surpasses fine-tuned OpenVLA and Magma on the SimplerEnv-Instruct benchmark, even without explicit textual reasoning (*Think*). (3) Explicit textual reasoning further enhances manipulation. Enabling thinking in the generalist model brings a 36.1% performance gain over direct instruction execution and even outperforms InstructVLA-expert paired with GPT-4o as an external interpreter.

## 6 Conclusion

We present InstructVLA, a unified VLA model that integrates multimodal reasoning and action generation. We further demonstrate how the embodied understanding ability can directly benefit the maipulation tasks. Our data and training pipeline enables leading performance across manipulation tasks, multimodal benchmarks, and real-world deployments, paving the way for more generalizable, interpretable, and interactive robots.

## REFERENCES

Figure AI. Helix, 2024. URL https://www.figure.ai/news/helix.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

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

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\backslash pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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 preprint arXiv:2212.06817*, 2022.

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

Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024a.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 2024b.

Zeren Chen, Zhelun Shi, Xiaoya Lu, Lehan He, Sucheng Qian, Hao Shu Fang, Zhenfei Yin, Wanli Ouyang, Jing Shao, Yu Qiao, et al. Rh20t-p: A primitive-level robotic dataset towards composable generalization agents. *arXiv preprint arXiv:2403.19622*, 2024c.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao,

Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025a.

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

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

M.J. Buehler E.L. Buehler. X-lora: Mixture of low-rank adapter experts, a flexible framework for large language models with applications in protein mechanics and design. 2024. URL https://arxiv.org/abs/2402.07148.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. URL https://arxiv.org/abs/2306.13394.

Ning Gao, Yilun Chen, Shuai Yang, Xinyi Chen, Yang Tian, Hao Li, Haifeng Huang, Hanqing Wang, Tai Wang, and Jiangmiao Pang. Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12187–12198, 2025.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22540–22550, 2025.

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, 2025. *URL https://arxiv. org/abs/2504.16054*, 1(2):3.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

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 preprint arXiv:2403.12945*, 2024.

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

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

Hao Li, Shuai Yang, Yilun Chen, Yang Tian, Xiaoda Yang, Xinyi Chen, Hanqing Wang, Tai Wang, Feng Zhao, Dahua Lin, et al. Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. *arXiv preprint arXiv:2506.19816*, 2025a.

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 preprint arXiv:2411.19650*, 2024a.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024b.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024c.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024d.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, et al. Hamster: Hierarchical action models for open-world robot manipulation. *arXiv preprint arXiv:2502.05485*, 2025b.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025c.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv e-prints*, pp. arXiv–2406, 2024c.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024d.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024e. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*, 2025.

Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. *arXiv preprint arXiv:2502.13142*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: latent codes as bridges in hierarchical robot control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8539–8546. IEEE, 2024.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U Tan, Deepanway Ghosal, Soujanya Poria, et al. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv preprint arXiv:2412.11974*, 2024.

Meituan LongCat Team, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen andf Zhang, Cheng Sun, et al. Longcat-flash technical report. *arXiv preprint arXiv:2509.01322*, 2025.

RealWorld Team. Realworldqa, 2024. URL `https://x.ai/news/grok-1.5v`.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in Neural Information Processing Systems*, 37:124420–124450, 2024a.

Lirui Wang, Jialiang Zhao, Yilun Du, Edward H Adelson, and Russ Tedrake. Poco: Policy composition from and for heterogeneous robot learning. *arXiv preprint arXiv:2402.02511*, 2024b.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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. *arXiv preprint arXiv:2502.05855*, 2025.

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. *arXiv preprint arXiv:2502.13130*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

James C Young, Rudy Arthur, and Hywel TP Williams. Cider: Context sensitive sentiment analysis for short-form text. *arXiv preprint arXiv:2307.07864*, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

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

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models, 2025. URL https://arxiv.org/abs/2501.10105.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025.

# Appendix

## CONTENTS

The supplementary material is organized as follows:

- Section A presents: (1) extended analysis of InstructVLA, (2) additional benchmarks on embodied understanding, (3) extra simulation benchmark and ablation study, (4) finetuning of OpenVLA under the same settings as InstructVLA, and (5) extra real-world ablation study.
- Section B discusses related concepts to InstructVLA and the proposed vision-language-action instruction tuning methods.
- Section C provides additional case analysis for InstructVLA, OpenVLA, and GPT-4o System2.
- Section D lists data annotation details, including GPT-4o prompt and dataset statistics. We further analyse the distribution of the instructions from two dimensions: task diversity and language diversity.
- Section E visualizes the SimplerEnv-Instruct benchmark and the acknowledgements of 3D assets.
- Section F details the model architecture, training configurations, inference speeds under different settings, and compute resources used.
- Section G shows several multimodal question answering examples.
- Section H describes the real-world experimental setup and provides example executions.
- Section I discusses the broader impacts, limitations, and outlines future directions for InstructVLA.

# A More Experiments and Analysis

## A.1 Further Discussions

Our further analysis is threefold. First, we present visualizations and scaling curves to examine the MoE and latent action designs. Second, we provide a detailed analysis of reasoning gains in manipulation tasks and case studies. Finally, we demonstrate that InstructVLA supports zero-shot dual-frequency generation to accelerate inference and compare the dataset scales used across different studies.

### A.1.1 Extra Model Design Analysis

The MoE and latent action are our key design components. We present an example illustrating the role of MoE under different task settings, including simple and reasoning instructions, with and without model reasoning. For latent action, we analyze its scaling behavior to guide future tuning.

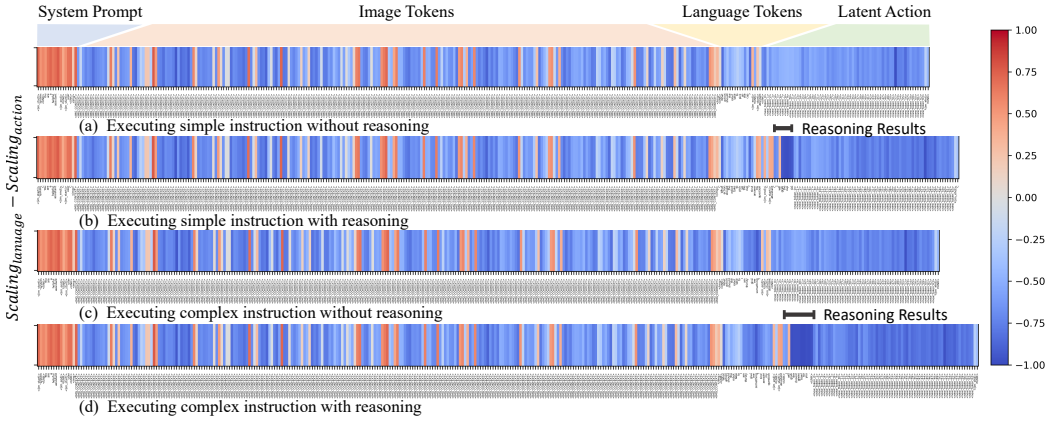



Figure 8: **Activation visualization.** We evaluate a WidowX zero-shot example across four settings. Red indicates stronger activation in the language adapter, while blue indicates stronger activation in the action adapter. The horizontal axis lists each language token. The generated tokens are marked.

**Analysis of MoE gating.** From the example in Figure 8, we draw the following intuitive conclusions:

- System prompts are primarily processed by the language adapter, reflecting its close connection to pretraining.
- Visual information is processed by both the language and action adapters, indicating that both semantic understanding and manipulation decision-making require visual inputs.
- During language generation, the model engages not only in multimodal reasoning but also in manipulation planning, as evidenced by the activation of the action expert. Notably, the action expert attends more strongly to nouns and verbs in the generated tokens, highlighting its role in instruction following.
- During latent action generation, the language expert plays a less prominent role. Instead, with multimodal reasoning, the model concentrates more effectively on action generation, as shown by the stronger activation of the action expert (deeper blue).

To conclude, the MoE has demonstrated its effectiveness in improving efficiency and handling heterogeneous datasets (Mu & Lin, 2025; E.L. Buehler, 2024; Zhou et al., 2022; Team et al., 2025; Liu et al., 2024a). In InstructVLA, we further investigate how the MoE facilitates interleaved multimodal reasoning and manipulation decision making.

**Effects of latent action.** Latent action tokens are a key design component for decoupling high-level VLM planning from low-level action generation. As shown in Figure 9, we vary the number of tokens from 16 to 128. Too few tokens limit behavioral diversity, while too many reduce training efficiency. A setting of 64 offers a good trade-off under our current configuration.

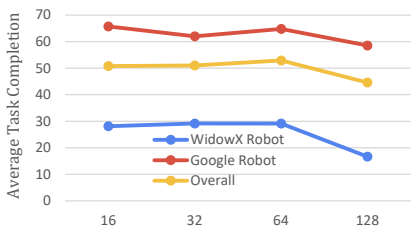



Figure 9: Impact of latent action token quantity on robot performance.

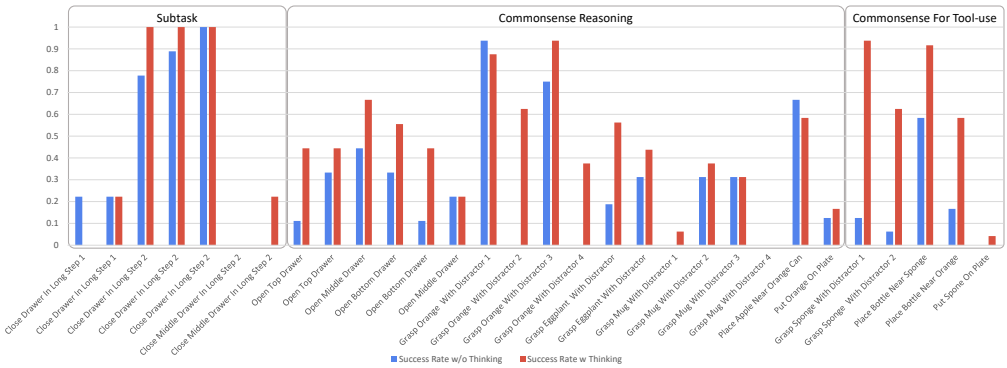



Figure 10: **Performance visualization** of 30 situated reasoning tasks with and without reasoning enabled. Activating reasoning in our generalist model generally improves performance. For clarity, tasks are grouped into three categories: *Subtask*, involving subtask identification; *Commonsense Reasoning*, requiring broad world knowledge; and *Commonsense for Tool Use*, focusing on tool-related reasoning.

### A.1.2 Extra Reasoning-Manipulation Analysis

In this section, we discuss the efficiency and design choices of VLA-IT training. We then analyze how multimodal reasoning benefits manipulation through fine-grained evaluation, examine its role in cross-embodiment generalization, and present a case study illustrating how a unique multimodal capability addresses challenging tasks.

**Effect of VLA-IT on Scaling and Reasoning.** As shown in Table 2, although the InstructVLA-expert model does not outperform the OpenVLA(OXE) on Situated Reasoning of SimplerEnv-Instruct, which benefits from direct full fine-tuning of the VLM backbone, InstructVLA-expert shows promising scaling ability in understanding complex instructions and performing test-time thinking after stage-2 VLA-IT training. This result reflects a deliberate design choice in InstructVLA, where latent action learning during pretraining focuses on querying from visual and simple instruction features rather than relying on the full semantic space of the VLM too early. This design offers two significant advantages. First, it preserves the original semantic space of the pretrained VLM, maintaining its vision-language capabilities. Second, it enables the model to integrate diverse reasoning contexts during VLA-IT training. These properties contribute to the strong performance gains achieved by our generalist model and demonstrate the effectiveness of this training paradigm.

**Embodied reasoning helps manipulation.** Allowing the model to perform test-time thinking by generating textual analysis of the given instruction can improve performance, particularly on situated reasoning tasks, as shown in Figure 11 (left). Notably, while the model with access to robot state outperforms the one without state when no instruction response is required, it provides limited performance gains when instruction following is involved. We hypothesize that state information helps the model retain manipulation skills but compromises its generalization to OOD environments and instructions.

**Fine-grained analysis of reasoning gains in manipulation tasks.** We compare the performance of the generalist model on SimplerEnv-Instruct with and without vision language reasoning, as

shown in Figure 10. A clear performance gap emerges in tasks involving commonsense tool use and interaction with articulated objects. This may result from instructions that do not explicitly state the intended actions and objects. For example, retrieving a cleaning tool from a drawer requires the robot to infer whether the prerequisite of an open drawer is satisfied, and to identify the sponge as the appropriate tool among several options. In addition to these cases, the reasoning process also improves performance on other situated reasoning tasks by grounding unfamiliar instructions using the pretrained in-domain knowledge of the vision language model.

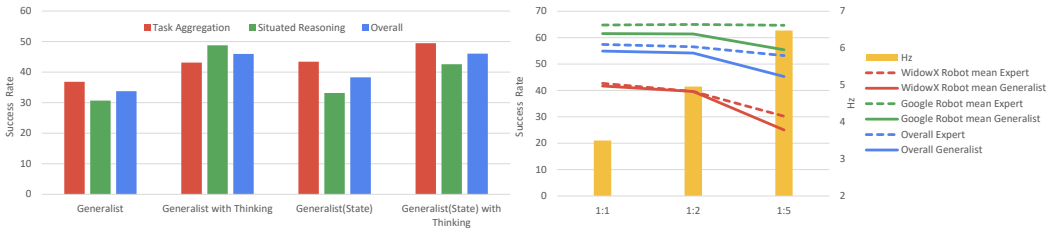



Figure 11: **Test-time tinking and dual-frequency evaluation.** "Expert" refers to the model after action pretraining, while "Generalist" denotes the model after VLA-IT tuning. For dual-frequency evaluation, the horizontal axis represents the ratio of VLM executions to expert model executions.

**VLA instruction tuning for cross-embodiment understanding.** To assess whether InstructVLA retains this capability, we evaluate three variants on SimplerEnv-Instruct (see Table 5): InstructVLA-Expert, trained solely on atomic instructions without test-time thinking; InstructVLA Generalist (Bridge), trained with the VLA-IT dataset on Bridge and the original Fractal dataset; and InstructVLA Generalist, trained with the full VLA-IT datasets across both environments. Adding the Bridge dataset results in a 139.4% improvement in Situated Reasoning performance for Generalist (Bridge) over the expert baseline, while task aggregation performance remains comparable. This discrepancy reflects differing generalization requirements: task aggregation emphasizes linguistic robustness, whereas Situated Reasoning demands vision-language grounding prior to action. The latter particularly benefits from the preserved reasoning capabilities of the pretrained VLM. As illustrated in Figure 12, the zero-shot model generates more diverse and accurate outputs than its fine-tuned counterpart.

Table 5: **Instruction tuning data ablation.** We evaluate three settings: without VLA-IT data, with data only on Bridge, and with VLA-IT data on both Fractal and Bridge. This ablation examines the contribution of the VLA-IT dataset and the cross-embodiment generalization of InstructVLA on SimplerEnv-Instruct.

| Instruction Tuning Data | | Name | Task Aggregation | Situated Reasoning | Overall |
|---|---|---|---|---|---|
| Bridge | Fractal | | | | |
| ✗ | ✗ | Expert | 20.8 | 10.4 | 15.6 |
| ✓ | ✗ | Generalist (Bridge) | 18.4 | 24.9 | 21.7 |
| ✓ | ✓ | Generalist | 43.3 | 48.8 | 46.0 |

**Case study on multimodal capability transfer.** As shown in Figure 13, we compare InstructVLA with OpenVLA (Kim et al., 2024), Magma (Yang et al., 2025), and CogACT (Li et al., 2024a), all using the same input (language instruction and a single image). InstructVLA-Expert, though trained without multimodal datasets, retains the OCR capability of the underlying VLM and achieves the best performance among baselines trained solely on manipulation data. Finetuning InstructVLA-Expert into InstructVLA-Generalist with multimodal and VLA-IT datasets further enhances performance. For autoregressive models such as OpenVLA and Magma, multimodal finetuning improves OCR ability. In contrast, CogACT, when fine-tuned from OpenVLA(OXE) only on manipulation data with an action head, shows improved in-domain performance (on SimplerEnv) but suffers in generalization.

### A.1.3 Extra Inference and Training Analysis

**Dual frequency inference.** To further analyze the relationship between latent actions generated by the VLM and the final decoded actions, we decouple the inference frequencies of the VLM and the

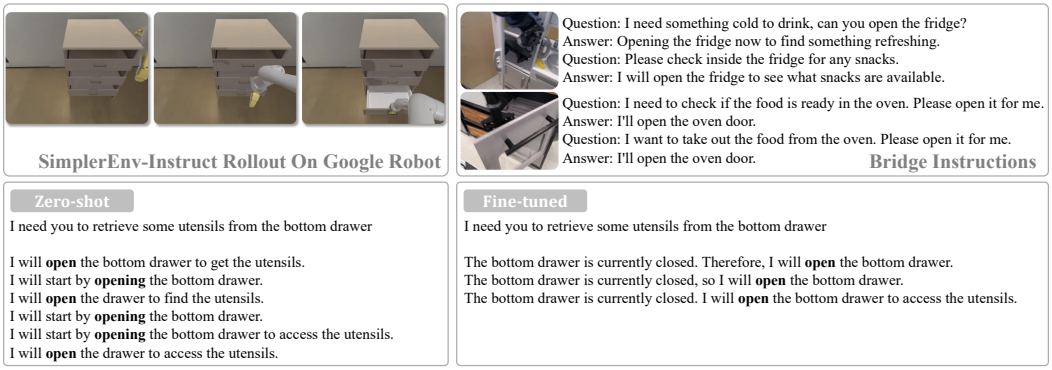



Figure 12: **Case study on cross-embodiment.** Top left: rollouts on SimplerEnv-Instruct. Top right: similar scenarios from the Bridge dataset with corresponding instructions. Bottom left: zero-shot results trained only on Bridge instructions. Bottom right: rollouts from the fine-tuned model.

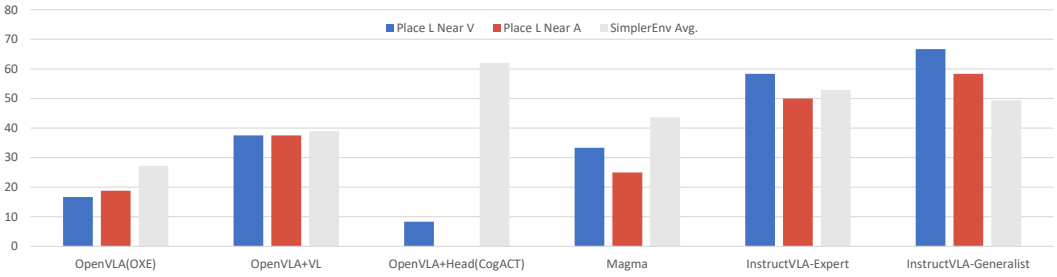



Figure 13: **Case study on multimodal capabilities.** OCR represents a unique multimodal skill of VLMs that is absent from typical manipulation datasets. We evaluate two tasks from the task aggregation set in SimplerEnv-Instruct, involving moving one letter to another (see Figure 18(1)). By comparing different finetuning paradigms, we assess how effectively multimodal capabilities are integrated into VLA models.

action expert, as illustrated in Figure 11 right. The results show that performance remains stable at a 1:2 ratio (VLM:expert), but begins to degrade at higher ratios. This suggests that latent actions offer relatively stable guidance to the action expert, reducing the need for frequent VLM queries.

**Training at scale.** A generalist VLA model with vision-language capabilities should be scalable across both manipulation and multimodal datasets. In this context, we compare datasets used by models claiming generalist abilities, as shown in Table 6. RoboMamba (Liu et al., 2024c) utilizes a limited manipulation dataset compared to other methods, while the dataset for ChatVLA (Zhou et al., 2025) is not reported. $\pi_{0.5}$ (Intelligence et al.) employs a significantly larger multimodal dataset than other approaches, though its multimodal performance is not disclosed. Magma uses more robot and multimodal data but achieves slightly worse performance on both multimodal and manipulation benchmarks compared to InstructVLA.

Table 6: **Data comparison of different methods.** "Trans." denotes transitions.

| | Magma(Yang et al., 2025) | ChatVLA(Zhou et al., 2025) | RoboMamba(Liu et al., 2024c) | $\pi_{0.5}$(Intelligence et al.) | InstructVLA |
|---|---|---|---|---|---|
| Manipulation Data | 9.4M Trans. | - | 10K Trans. | 400 Hours | 469 Hours/ 5.9M Trans. |
| Multimodal Data | 1.2M Images + 4M Videos | 54K | 1.5M | >7M | 2M |

### A.2 EMBODIED UNDERSTANDING EVALUATION

Table 7: **VLA-IT captioning evaluation.** "Sentence-BERT" and "SimCSE" represent learning-based evaluation methods, while the remaining metrics are traditional n-gram-based evaluations focused on word distribution.

| Methods | # Params | Sentence-BERT | SimCSE | BLEU-1 | BLEU-4 | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| Qwen2-VL (Wang et al., 2024c) | 1.5B | 61.3 | 67.5 | 16.8 | 1.5 | 12.4 | 0.30 |
| GPT4o (OpenAI, 2023) | - | 60.7 | 67.1 | 16.3 | 1.8 | 16.2 | 0.09 |
| OpenVLA(VLA-IT) (Kim et al., 2024) | 7B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 |
| Magma (Yang et al., 2025) | 8B | 59.8 | 66.7 | 12.4 | 1.2 | 12.3 | 0.12 |
| InstructVLA(Generalist) | 1.5B | 72.0 | 77.0 | 44.3 | 8.2 | 18.7 | 0.84 |

Table 8: **VLA-IT question-answering evaluation.**

| Methods | # Params | Sentence-BERT | SimCSE | BLEU-1 | BLEU-4 | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| Qwen2-VL (Wang et al., 2024c) | 1.5B | 51.9 | 53.4 | 15.3 | 2.8 | 17.9 | 0.82 |
| GPT4o (OpenAI, 2023) | - | 63.6 | 63.6 | 29.6 | 19.9 | 9.8 | 1.16 |
| OpenVLA(VLA-IT) (Kim et al., 2024) | 7B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 |
| Magma (Yang et al., 2025) | 8B | 53.5 | 54.5 | 23.7 | 5.7 | 21.6 | 1.04 |
| InstructVLA(Generalist) | 1.5B | 64.9 | 65.9 | 44.6 | 17.4 | 23.5 | 1.85 |

Table 9: **VLA-IT instruction response evaluation.** We use "context creation" annotations, as they present a more challenging and diverse set of instructions.

| Methods | # Params | Sentence-BERT | SimCSE | BLEU-1 | BLEU-4 | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| Qwen2-VL (Wang et al., 2024c) | 1.5B | 52.3 | 54.0 | 5.6 | 1.5 | 11.6 | 0.09 |
| GPT4o (OpenAI, 2023) | - | 52.8 | 54.1 | 17.8 | 4.2 | 20.6 | 1.02 |
| OpenVLA(VLA-IT) (Kim et al., 2024) | 7B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 |
| Magma (Yang et al., 2025) | 8B | 10.9 | 13.6 | 3.7 | 0.8 | 1.6 | 0.00 |
| InstructVLA(Generalist) | 1.5B | 71.6 | 73.1 | 50.2 | 24.1 | 25.8 | 2.26 |

In addition to the multimodal and closed-loop evaluations presented in the main results, we conduct supplementary language evaluations on the proposed VLA-IT dataset. This evaluation uses manually verified VLA-IT annotations on the Bridge dataset (Ebert et al., 2021), chosen for its diversity and distinct validation split. We generate 1,000 annotations following the method described in the VLA-IT dataset generation section. Two evaluation metrics are employed: (1) learning-based methods (Reimers, 2019; Gao et al., 2021), and (2) traditional metrics (Papineni et al., 2002; Young et al., 2023; Banerjee & Lavie, 2005).

The captioning, question-answering and instruction-following results are presented in Tables 7 to 9. We select Qwen2-VL (Wang et al., 2024c) and GPT-4o (OpenAI, 2023) as zero-shot VLM baselines, and include Magma (Yang et al., 2025) (zero-shot) and OpenVLA (Kim et al., 2024) fine-tuned on the VLA-IT dataset as baselines for VLA models.

Although OpenVLA is fine-tuned on the VLA-IT dataset, it fails to generate complete sentences under the same evaluation setting as InstructVLA, despite the performance on multiple-choice benchmarks reported in our main results. This suggests a significant loss of its free-form dialogue capability. Magma performs well on question answering and captioning tasks. However, it struggles with instruction response (Figure 14), often generating outputs misaligned with the given image. *We hypothesize that this failure stems from the similarity between these instructions and the atomic commands used in finetuning manipulation datasets, which disrupts the coherence of the language latent space near the action latent space.* This suggests a limited capacity to interpret and generalize free-form instructions, hindering effective transfer of vision-language capabilities.

InstructVLA achieves state-of-the-art performance, while GPT4o demonstrates competitive results. We visualize three episodes in Figure 15. GPT-4o generates more detailed captions but occasionally exhibits minor hallucinations. In the instruction response task, InstructVLA produces clearer and more grounded responses compared to GPT-4o, benefiting from the integration of ground-truth atomic instructions during the data annotation process, as discussed in Section D.4.

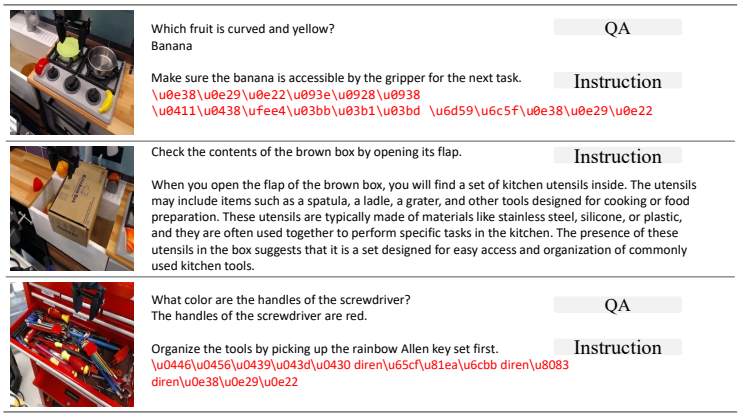



Figure 14: **Magma results.** Magma's responses collapse when given instructions resembling those in its manipulation tasks, possibly due to learned actions interfering with its language latent space.

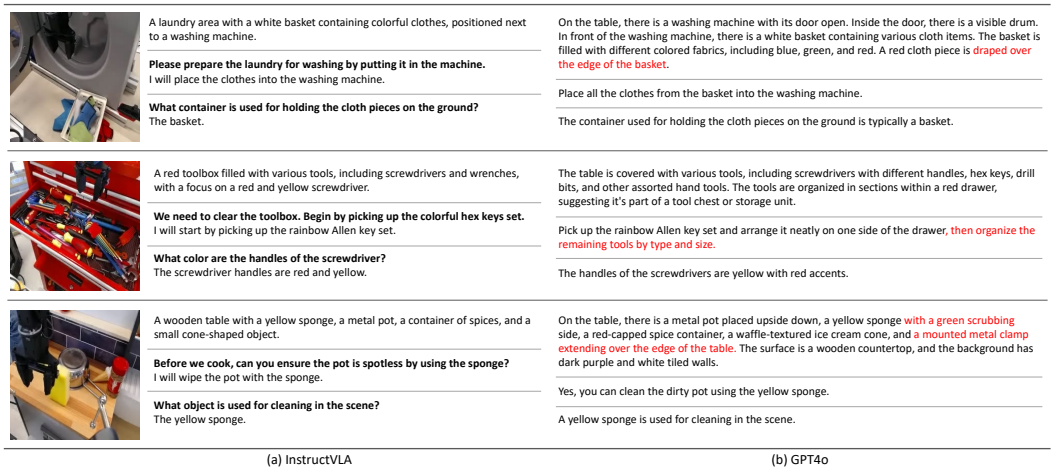



Figure 15: **Comparison with GPT-4o.** We visualize three examples from the VLA-IT language validation set. Each example includes a scenario caption (top), instruction response (middle), and question answering (bottom). The GPT-4o column displays **responses only**, as the instructions are identical across models.

### A.3 Extra Manipulation Benchmark

Table 10: **LIBERO benchmark results.** We present the success rate and standard error for each method across four task suites, which are averaged over three random seeds with 500 trials. "KI" denotes knowledge insulating from(Driess et al., 2025).

| | Spatial | Object | Goal | 10 (Long) | Average |
|---|---|---|---|---|---|
| OpenVLA-7B (Kim et al., 2024) | $84.7 \pm 0.9$ | $88.4 \pm 0.8$ | $79.2 \pm 1.0$ | $53.7 \pm 1.3$ | $76.5 \pm 0.6$ |
| OpenVLA-OFT-7B (Kim et al., 2025) | $97.6 \pm 0.9$ | $98.4 \pm 0.8$ | $97.9 \pm 1.0$ | $94.5 \pm 1.3$ | $97.1 \pm 0.6$ |
| SpatialVLA-2B (Qu et al., 2025) | $88.2 \pm 0.5$ | $89.9 \pm 0.7$ | $78.6 \pm 0.6$ | $55.5 \pm 1.0$ | $78.1 \pm 0.7$ |
| $\pi_0$-2B (Black et al., 2024) | $96.8 \pm 0.8$ | $98.8 \pm 0.9$ | $95.8 \pm 1.1$ | $85.2 \pm 1.2$ | $94.2 \pm 0.9$ |
| $\pi_0$-FAST-2B (Pertsch et al., 2025) | $96.4 \pm 0.7$ | $96.8 \pm 0.7$ | $88.6 \pm 1.0$ | $60.2 \pm 1.4$ | $85.5 \pm 1.0$ |
| GR00T-N1-1.34B (Bjorck et al., 2025) | $94.4 \pm 0.9$ | $97.6 \pm 1.0$ | $93.0 \pm 1.2$ | $90.6 \pm 1.0$ | $93.9 \pm 1.1$ |
| $\pi_{0.5}$ + KI (from scratch) (Intelligence et al.) | 96.6 | 97.2 | 94.6 | 84.8 | 93.3 |
| $\pi_{0.5}$ + KI (from generalist model) (Intelligence et al.) | 98.0 | 97.8 | 95.6 | 85.8 | 94.3 |
| DexVLA-1.5B Wen et al. (2025) | 97.2 | 99.1 | 95.6 | - | - |
| InstructVLA (w/o wrist view) | 92.4 | 95.6 | 92.0 | 76.6 | 89.2 |
| InstructVLA-1.5B | $97.3 \pm 0.5$ | $99.6 \pm 0.0$ | $96.5 \pm 0.5$ | $89.8 \pm 1.6$ | $95.8 \pm 0.4$ |

**Benchmarks and baselines.** We evaluate InstructVLA on the LIBERO simulation benchmark (Liu et al., 2024b), which includes diverse robotic manipulation tasks in simulated environments. Following OpenVLA (Kim et al., 2024), we conduct experiments on four task suites, each containing 10 tasks with 50 human-teleoperated demonstrations. These suites assess spatial reasoning (LIBERO-Spatial), object type understanding (LIBERO-Object), task-oriented behaviors (LIBERO-Goal), and generalization to long-horizon tasks involving diverse objects, layouts, and goals (LIBERO-Long).

Our baselines fall into two categories: (i) generalist manipulation policies, including OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025), SpatialVLA (Qu et al., 2025), $\pi_0$(Black et al., 2024), and $\pi_0$-FAST(Pertsch et al., 2025); and (ii) manipulation policies with multimodal ability, including GR00T-N1 (Bjorck et al., 2025), DexVLA Wen et al. (2025) and $\pi_{0.5}$(Intelligence et al.) with knowledge insulation(Driess et al., 2025).

**Training details.** We augment InstructVLA with wrist-view images from the LIBERO training set (Liu et al., 2024b). Specifically, both the main and wrist-view images are provided to the VLM and the action expert. To reduce the tokenized input length, the two images are concatenated and resized into a single frame for VLM. Training follows the same hyperparameters as the Simpler-Env experiments and is performed on a single A800 node with 8 GPUs using a global batch size of 256, with evaluation every 1.5K steps.

**Results.** As shown in Table 10, InstructVLA achieves competitive performance despite not being pretrained on large-scale manipulation datasets like $\pi_{0.5}$(Intelligence et al.; Driess et al., 2025) and using a much smaller VLM backbone than OpenVLA-OFT(Kim et al., 2025). Compared with recent VLAs such as DexVLA Wen et al. (2025), InstructVLA attains higher performance with a substantially smaller action model (134M versus 1B).

### A.4 Data Ablation on OpenVLA

Table 11: **Data ablation on OpenVLA.** "+VL" indicates finetuning OpenVLA with the same multimodal dataset used by InstructVLA. "+VLA-IT" refers to finetuning OpenVLA with the same VLA-IT dataset as InstructVLA. "+GPT4o" denotes using GPT4o as system 2 to translate free-form instructions into atomic ones.

| | OpenVLA (OXE) | OpenVLA + VL | OpenVLA + VL + VLA-IT | OpenVLA + VL + GPT4o | InstructVLA |
|---|---|---|---|---|---|
| task aggregation | 14.8 | 28.3 | 30.5 | 38.8 | 43.3 |
| Situated Reasoning | 13.6 | 19.5 | 17.4 | 32.4 | 48.8 |
| Average | 14.2 | 23.9 | 24.0 | 35.6 | 46.0 |

To investigate whether the performance gain of VLA-IT arises solely from the dataset itself, we reimplement the training procedure of the InstructVLA on OpenVLA (Kim et al., 2024), which represents a class of models trained under the action-only paradigm. As shown in Table 11, OpenVLA benefits from both vision-language and VLA instruction tuning data, with the latter showing greater

improvement in the task aggregation setting. This is attributed to exposure to more diverse instructions. However, performance on the situated reasoning setting remains unchanged, likely due to catastrophic forgetting caused by the action-only training paradigm, which limits OpenVLA's ability to leverage the VLM's reasoning ability through simple finetuning.

The greatest performance gain is observed when GPT-4o is introduced as an auxiliary System 2 in both evaluation settings. However, overall performance remains inferior to InstructVLA, as GPT-4o cannot fully ground free-form instructions to the atomic skills on which OpenVLA is pretrained.

### A.5 Real-world Ablation

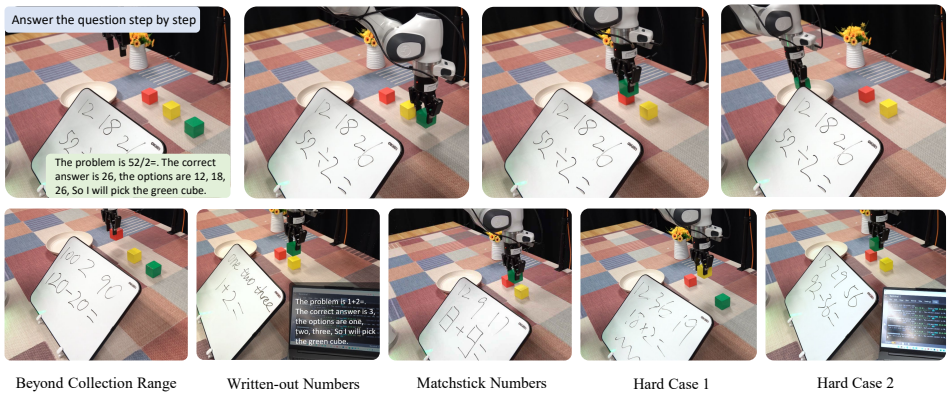



Figure 16: **Real-world ablation study.** The first row depicts the reasoning responses and the rolled-out actions, while the second row illustrates five categories of generalization.

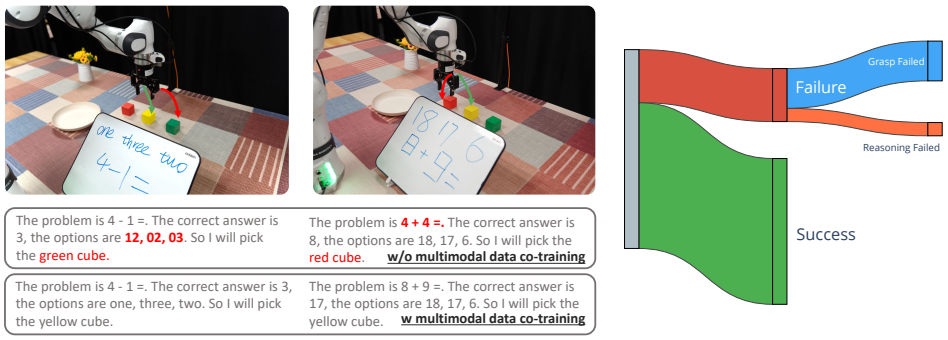



Figure 17: **Reasoning examples.** Two evaluation cases are presented to illustrate the role of multimodal datasets. We further summarize the results of InstructVLA in a Sankey diagram.

**Setup.** This case study evaluates the role of multimodal datasets in manipulation tasks. The robot setup follows our few-shot Frank evaluation. As shown in Figures 16 and 17, the model must first perform OCR to recognize the formula on the board and its answer options, then compute the result, and finally control the robot to grasp the correct object. This task mirrors a shopping scenario where robots often need to read prices and perform simple calculations to satisfy a requirement. The study jointly assesses OCR and calculation abilities, which are expected to benefit from multimodal data. To reduce bias, each case is evaluated three times with different target objects. In total, 250 training cases are collected but excluded from evaluation.

The in-domain tasks are defined as calculations within the range of the training data and written in a similar format. Generalization tasks are divided into five types: (1) Beyond Collection Range, (2) Written-out Numbers, (3) Matchstick Numbers, (4) Hard Case 1 (digits partially occluded with superimposed lines), and (5) Hard Case 2 (involving more complex calculations).

**Analysis.** By co-training with a general multimodal dataset, we observe that InstructVLA performs better on the tasks of *Written-out Numbers*, *Matchstick Numbers*, and *Hard Case 1*. We attribute this improvement to the inclusion of general OCR data within the multimodal dataset. Although the multimodal dataset is unfiltered (i.e., identical to the corpus used for training a VLM such as Bunny), it nonetheless enhances the instruction generalization for these specific tasks.

The SOTA VLA $\pi_0$ (Black et al., 2024), although pretrained on DROID Khazatsky et al. (2024), however, produces near-random results: although each grasp is executed precisely, the model frequently selects the wrong target object. Interestingly, when the third-view camera, which capturing the board with expressions and options, is masked, $\pi_0$ still behaves similarly. This suggests that $\pi_0$ largely ignores reasoning cues and overfits to the wrist view. While it performs precise grasping, the overall outcomes remain unsatisfactory.

# B Extra Related Works

In this section, we delineate the distinctions between InstructVLA and several similarly named methods that differ substantially in their conceptual foundations and objectives.

## B.1 Embodied Instruction Tuning

**Vision-Action Instruction Tuning.** The concept of Vision-Action Instruction Tuning is introduced in LLARVA Niu et al. (2024), which unifies robotic tasks through structured prompts and 2D trace supervision for cross-embodiment pretraining. In contrast, InstructVLA extends this idea by focusing on preserving the multimodal knowledge of VLMs and bridging high-level human instructions with low-level manipulation skills, enabling generalization to diverse tasks that require common-sense reasoning.

**Visuomotor Instruction Tuning.** The concept of Visuomotor Instruction Tuning is purposed in LLaRA Li et al. (2024b). This approach formulates robot policies as visuo-textual conversations and produces 2D keypoints and rotations for manipulation. However, it functions primarily as a high-level planner, and its outputs require additional adaptation before being directly executed on robots.

## B.2 Multi-stage Training

**OpenVLA-OFT.** OpenVLA-OFT (Kim et al., 2025) extends OpenVLA (Kim et al., 2024) by incorporating FiLM layers, Parallel decoding, MLP action head, and has been applied to fine-tuning on smaller simulation datasets such as LIBERO Liu et al. (2024b). This approach demonstrates the effectiveness of architectural enhancements for improving manipulation performance in specific domains. However, while these techniques improve in-domain performance, they fall short in reasoning-centric settings such as SimplerEnv-Instruct, as shown in Figure 6 (b). In contrast, our work moves beyond architectural modifications by emphasizing generalizable manipulation with textual reasoning through MoE adaptation, latent action methods, and a comprehensive data and evaluation pipeline. With the proposed VLA-IT training paradigm, our generalist model achieves nearly a $2\times$ improvement over models that rely solely on architectural designs.

**Embodied Chain-of-Thought.** ECoT (Zawalski et al., 2024) introduces chain-of-thought (CoT) supervision to link reasoning with manipulation and follows a standard "pretrain-then-instruction-tune" paradigm. However, it relies on full-model pretraining fine-tuning, as in OpenVLA (Kim et al., 2024), which leads to catastrophic forgetting of vision-language capabilities. *In contrast, InstructVLA adopts a two-stage design: the first stage injects action-generation ability while deliberately preserving the multimodal knowledge of the pretrained VLM.* This approach ensures that the model retains open-world understanding and general multimodal reasoning, both of which are largely lost in ECoT. The second stage then strengthens multimodal reasoning and manipulation alignment. Consequently, InstructVLA supports broader inference modes (reasoning + manipulation, direct manipulation, and multimodal VQA) and achieves stronger performance with substantially fewer trainable parameters.

# C CASE STUDY

## C.1 REASONING CASES IN SIMPLERENV-INSTRUCT

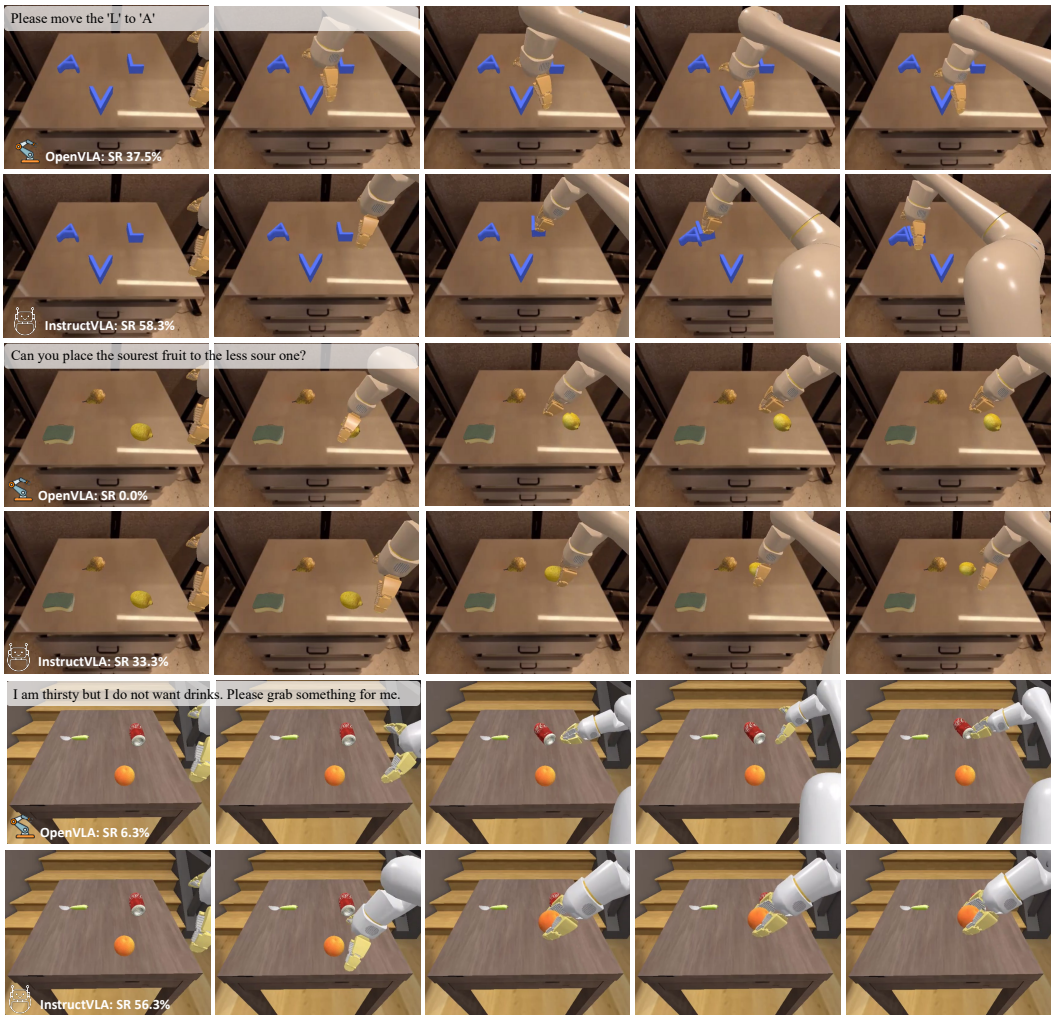



Figure 18: **Reasoning cases in SimplerEnv-Instruct.** Three cases of the VL fine-tuned OpenVLA and InstructVLA-Generalist. "SR" denotes success rate.

We present three representative reasoning cases in Figure 18. In the first example, OpenVLA fails to associate the letters "V" and "L" with their corresponding shapes in the image, resulting in consistent failure to grasp in all similar scenarios. In the second case, OpenVLA does not correctly associate the concept of "sour" with the corresponding fruit. As a result, its action is influenced by both the pear and lemon, leading to a grasp attempt between them that strikes the table. In the final example, OpenVLA fails to interpret the negation in the instruction and incorrectly grasps Coke instead of orange.

### C.2 FAILURE CASES

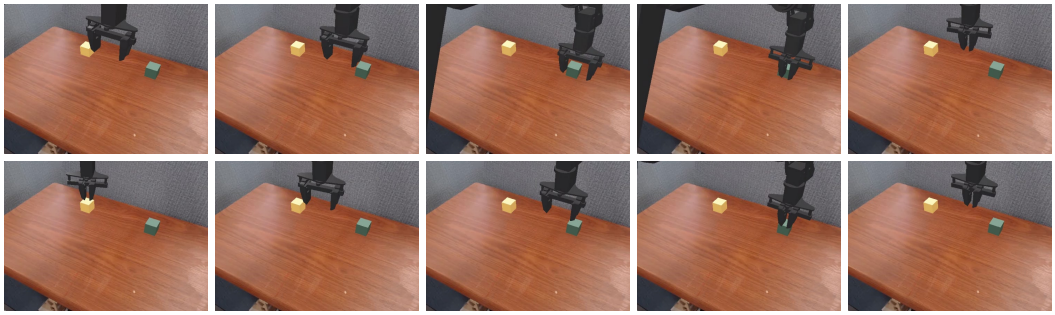


Figure 19: **Failure case 1 of InstructVLA.** The model receives only a third-person view image as visual input, making it difficult to estimate depth or the gripper's relative position to the object. Consequently, it fails to grasp the object accurately, despite the gripper appearing aligned with the target in the image.

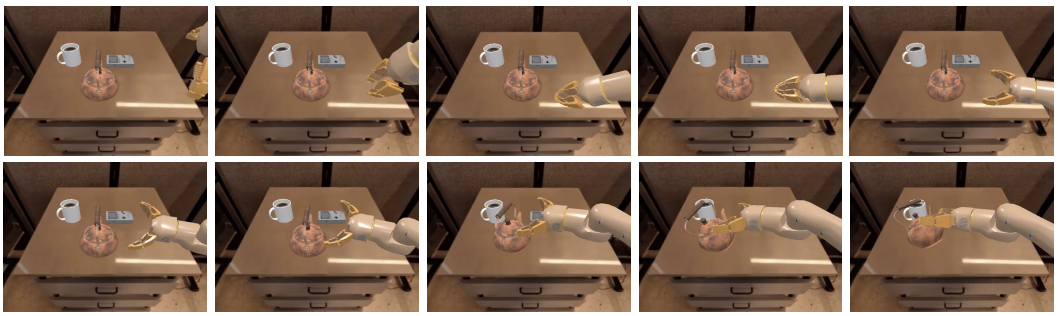


Figure 20: **Failure case 2 of InstructVLA.** The model fails to accurately estimate depth due to the real-to-sim gap, specifically the absence of arm reflection on the table, which causes the robot to become stuck in an out-of-distribution position.

We illustrate two representative failure cases of InstructVLA in Figures 19 and 20. While some failures may result from the real-to-sim gap, incorporating additional sensory inputs such as depth information and robot state may enhance performance. We leave this exploration for future work. Additionally, we observe that the model achieves higher success rates in language responses than in action execution, suggesting that multimodal understanding is more readily transferable than manipulation skills. This highlights a fundamental challenge in the development of embodied models.

### C.3 GPT4o AS THE AUXILIARY SYSTEM 2

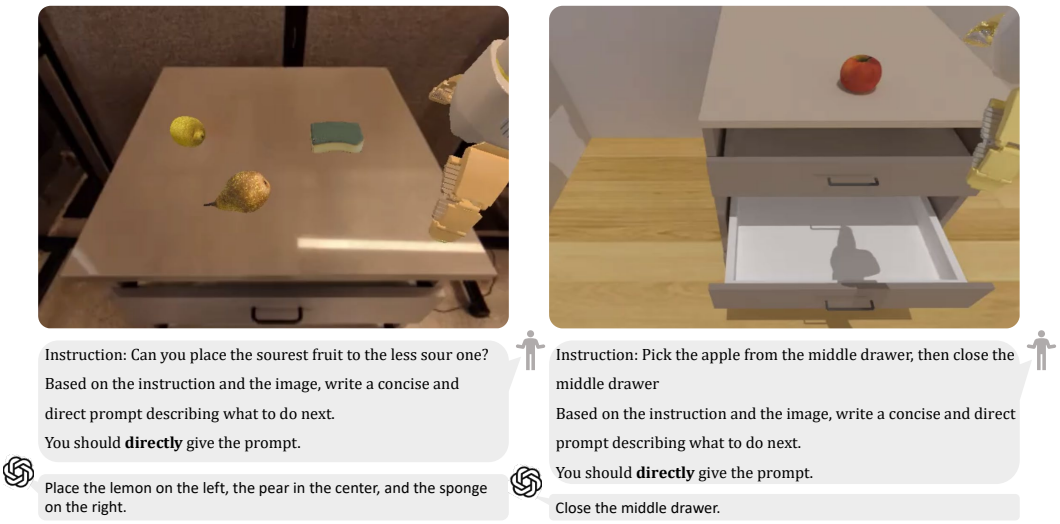



Figure 21: **GPT-4o as the auxiliary system 2.** We prompt GPT-4o with the first image from the environment along with the instruction, asking it to rewrite the prompt in a simple and clear format.

A strong baseline for InstructVLA integrates an expert model capable of executing atomic instructions with GPT-4o as an instruction parser to decompose complex, free-form commands for decision-making (Shi et al., 2025; Gao et al., 2025). The prompt used is listed in Prompt 1, and it was evaluated and refined on 20 test cases from the task aggregation to ensure reliable performance. Results on additional test cases are presented in Figure 21. GPT-4o successfully identified the atomic instruction in the second case but failed in the first.

During evaluation, GPT-4o is invoked only in the initial step to ensure an unobstructed view of the scene and to generate a free-form instruction. We do not provide a closed set of task-relevant instructions for selection, as the training set (Figure 22) lacks sufficient diversity in instructions and objects, and therefore does not adequately cover the evaluation settings. Across 80 evaluation cases, GPT-4o frequently fails in physical grounding, maintaining coherence, and accurately interpreting the scene.

GPT-4o System-2 Prompt

Instruction: Can you place the sourest fruit to the least sour one?
Based on the instruction and the image, write a concise and direct prompt to describe what to do next.
You should **directly** give the prompt.

# D Data Annotation Details and Analysis

The data analysis and GPT4o prompt are listed as follows (Figure 22 and Prompt 2).

## D.1 Language motion pre-training data

Language motion (Belkhale et al., 2024) provides intuitive linguistic descriptions of basic end-effector movements, which can be distilled into latent actions. We compute the relative movement of the end-effector between the $t$-th and $(t + W)$-th steps, using a window size $W$. The final labels are formatted, such as "move right and open the gripper".

## D.2 Task Diversity Analysis

We categorize tasks into two broad classes: **Command Rewriting / Context Creation** and **Question Answering**. Each class includes several common task types:

### Command Rewriting / Context Creation

- **Complex Object Referencing:** Uses attributes, pronouns, or relational terms to reference an object.
  *Example:* "Place the red item next to the box."
- **Novel Action Referencing:** Rephrases a previously known action using a different verb or motion.
  *Example:* "Shut the drawer" (instead of "Close the drawer").
- **Negative Task Specification:** Specifies the correct action by negating incorrect alternatives.
  *Example:* "I'm thirsty, but I don't want sparkling water—bring me something else."
- **Subtask Identification:** Isolates a step from a multi-step instruction with a clear sequential order.
  *Example:* From "Take the spoon out of the top drawer," execute only the first step.
- **Situated Task Identification:** Infers the required action based on contextual cues or situational conditions.
  *Example:* "I want to clean the table. What should I use?"
- **Direct Instruction:** Provides an explicit and unambiguous command.
  *Example:* "Organize the drinks by putting the green can next to the Coke can."
- **Tool-Use Understanding:** Refers to an object by its utility or function rather than its name.
  *Example:* "Hand me something to cut with" (instead of "Use the knife").

### Question Answering

- **Quantitative Identification:** Requires determining the number or quantity of items.
  *Example:* "How many apples are on the table?"
- **Spatial Identification:** Involves spatial relationships between objects or with the user.
  *Example:* "Is the cup on the left or the right of the plate?"
- **Visual Identification:** Focuses on appearance-based attributes such as color or shape.
  *Example:* "Which one is the metallic silver object?"
- **Commonsense Answering:** Requires everyday reasoning or world knowledge.
  *Example:* "Which of these would you use to cut paper?"
- **State Identification:** Determines the current condition or status of an object.
  *Example:* "Is the drawer currently open or closed?"

The data examples for VIA-IT are provided in Figures 23 and 24.

## D.3 Prompting

The Prompt 2, along with three images captured at the beginning, middle, and end of each episode, is packaged and sent to GPT-4o. Episodes from the Bridge dataset (Ebert et al., 2021) that lack valid instructions are excluded from annotation.

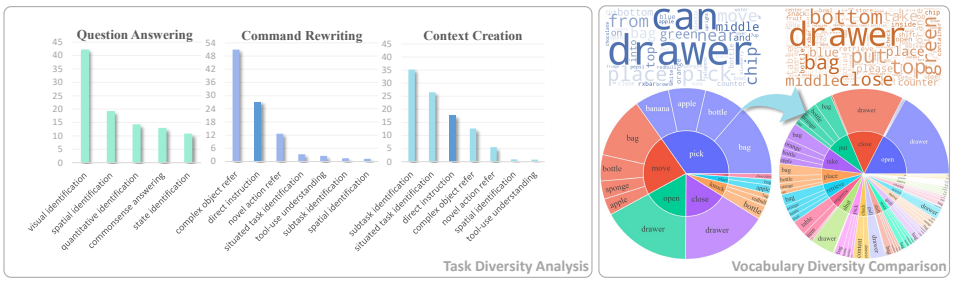



Figure 22: **Data analysis.** Left: We manually identify common task categories and calculate the distribution. The proportion of direct prompts is reduced in favor of more diverse, free-form expressions. Right: Word cloud and verb-noun analyses compare the original Fractal instructions with the VLA-IT corpus.

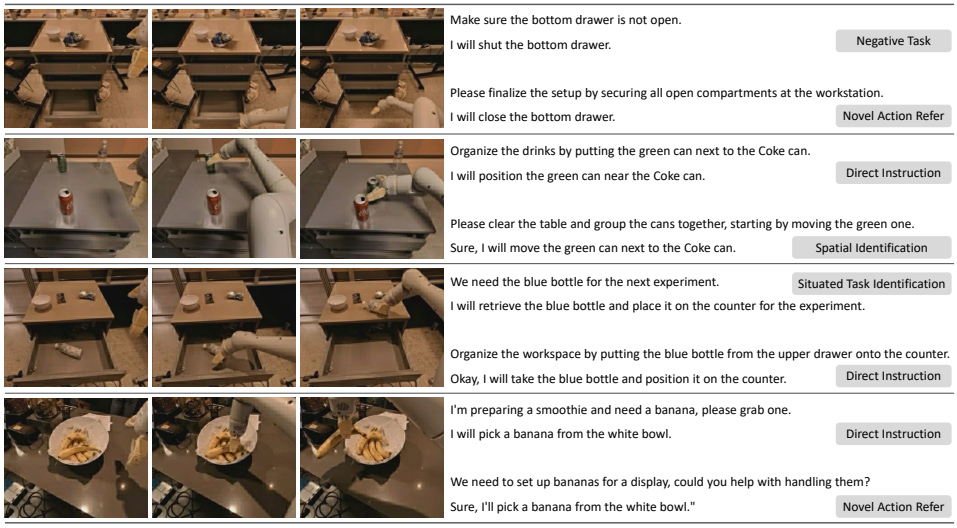



Figure 23: More VLA instructions on Fractal dataset.

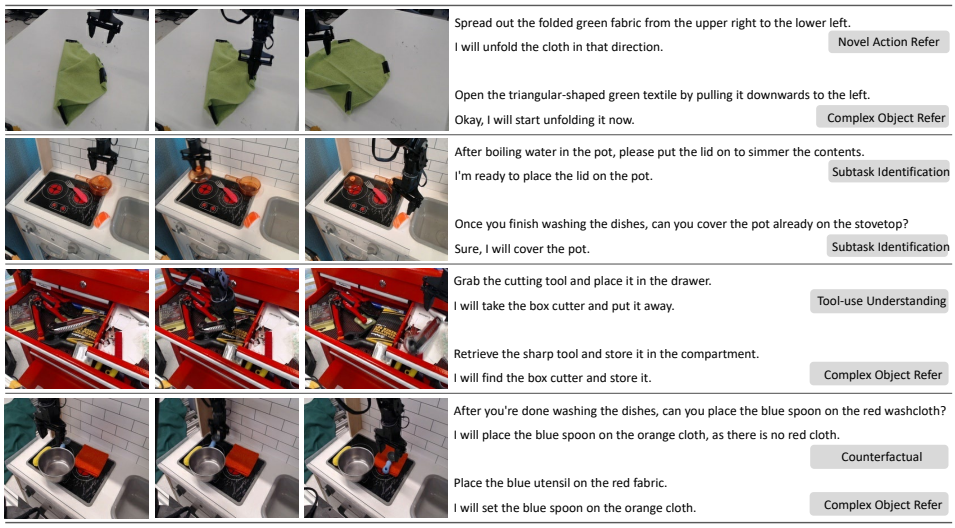



Figure 24: More VLA instructions on Bridge dataset.

Data Annotation Prompt

Imagine a robot assistant operating in a laboratory or household environment. The robot is expected to follow diverse commands based on realistic tasks and human interactions. Your task is to:

1. Write a caption to describe the visual scene shown in the **first image**. You should **NOT** include the robot itself here.
2. Based on the given robot task description and the images, generate new user instructions and corresponding robot responses with QA pairs.

The new user instructions should align with the actions performed by the robot in the images and with the environment shown in the images. You are required to produce three categories of instructions:

1. **Command Rewriting (CR)**: Rephrase the task description using diverse language styles and vocabulary. You may refer to objects by their utility, color, shape, or other attributes, but ensure the attribute you use is unique to each object.
2. **Context Creation (CC)**: Generate detailed scenarios where the robot needs to perform the given instruction. The situation should involve realistic surroundings or tasks where this instruction would be necessary. You may also simulate a long-horizon task based on the context provided by the image. Your generated question should **NOT** include the answer itself.
3. **Scene-related Commonsense QA (QA)**: Generate some other QA pairs that are related to the scene. The answer should be concise and consistent among the three images.

For each instruction, provide a concise robot response that clearly (use simple words) communicates the next action the robot will take. **Do not chain multiple actions together using phrases like "and then."** If necessary, the response may include a brief explanation of the reasoning. Avoid repeating the instruction in the response.
**Response Format**: You MUST respond in JSON format. You should include `"Caption"`, `"CR"`, `"CC"`, and `"QA"` in your response. You should create 1-3 entries for each of CR, CC, and QA.
**Example 1**: For the instruction "Close middle drawer":
*(Corresponding three images omitted)*
**Caption**: "A table with a Coke and chips on top, with its middle drawer open."

```
{
  "Caption": "A table with a Coke and chips on top, with its middle drawer open.",
  "CR": [ { "question": "Push the middle drawer closed.",
            "answer": "Ok, I will close it." },
          { "question": "Ensure the center drawer is closed.",
            "answer": "I will close the drawer." } ],
  "CC": [ { "question": "I want you to take out the Coke from the middle drawer and closing it.",
            "answer": "The Coke is on the table, and the middle drawer is empty. So, I should close the middle drawer." },
          { "question": "Please push the middle drawer shut so we can clear the workspace.",
            "answer": "Okay, I will close the middle drawer." } ],
  "QA": [ { "question": "What is in the middle drawer?",
            "answer": "The middle drawer is empty." },
          { "question": "How many Coke cans are on the table?",
            "answer": "One." } ]
}
```

**Example 2**: For the instruction "move the apple near the Coke":
*(Corresponding three images omitted)*
**Caption**: "A table with Coke, apple, and soap on it."

```
{
  "Caption": "A table with Coke, apple, and soap on it.",
  "CR": [ { "question": "Move the healthy food near the Coke.",
            "answer": "The healthy food refers to the apple, and I will move the apple to the Coke." },
          { "question": "Move the apple to the cylindrical-shaped object.",
            "answer": "Of course!" } ],
  "CC": [ { "question": "Gather all objects near the Coke, except the soap.",
            "answer": "I will move the apple to the Coke." } ],
  "QA": [ { "question": "I'm thirsty, what can I have?",
            "answer": "The Coke is on the table." },
          { "question": "What is the healthy food on the table?",
            "answer": "The apple." } ]
}
```

Your task description is "`<placeholder>`".
Now give your response in JSON format.

### D.4 Ground Truth Instruction for Data annotation

During data generation, we observe that GPT-4o often struggles to accurately interpret robot behavior using only the three provided images, performing noticeably worse than humans. To quantify this, we randomly sample 100 examples and prompt GPT-4o to generate our four types of annotations using a similar prompt (excluding the ground truth instruction from a human expert). We then manually evaluate the correctness of the results: a sample is scored as 1 if no obvious errors are found, 0.5 if minor errors are present, and 0 if completely incorrect.

The results are summarized in Tables 12 and 13, with two representative cases illustrated in Figures 26 and 27. In the first case, GPT-4o hallucinates the robotic arm as a bread roll, leading to an incorrect caption and instruction. In the second, it reverses the temporal order of actions, resulting in an inaccurate annotation.

We attribute this performance gap to GPT-4o's lack of temporal grounding and the low visual quality of images in manipulation datasets. In contrast, human-provided instructions inherently encode temporal links across the image sequence by grounding the task in context, identifying target objects, and specifying corresponding robot actions. This finding underscores that, despite their impressive capabilities, even state-of-the-art VLMs lack embodied experience and temporal grounding, limiting their ability to infer fine-grained actions in robot manipulation tasks.

Table 12: **Data annotation success rate.** GPT-4o shows a significant performance drop without ground truth instructions during data annotation.

| Method | Success Rate |
|---|---|
| With GT Instruction | 95.4% |
| Without GT Instruction | 45.0% |

Table 13: **Distribution of common error types.** Error analysis of GPT-4o annotations generated without access to ground truth instructions, with long-tail errors omitted.

| **Error Type** | **Percentage** |
|---|---|
| Ignore Vision Context | 32.5% |
| Reverse Temporal Order | 10.2% |
| Minor Object Hallucination | 5.7% |

### D.5 Language Motion Examples

Language motion (Belkhale et al., 2024) describes end-effector movements using natural language, enhancing the VLM's understanding of robotic manipulation. To generate such annotations, we leverage proprioceptive data that captures the end-effector's position and orientation relative to the robot base. While the Bridge dataset (Ebert et al., 2021) adopts annotations from ECoT (Zawalski et al., 2024), we additionally annotate the Fractal dataset (Brohan et al., 2022) using a similar approach. The examples on the Fractal dataset are presented in Figure 25.

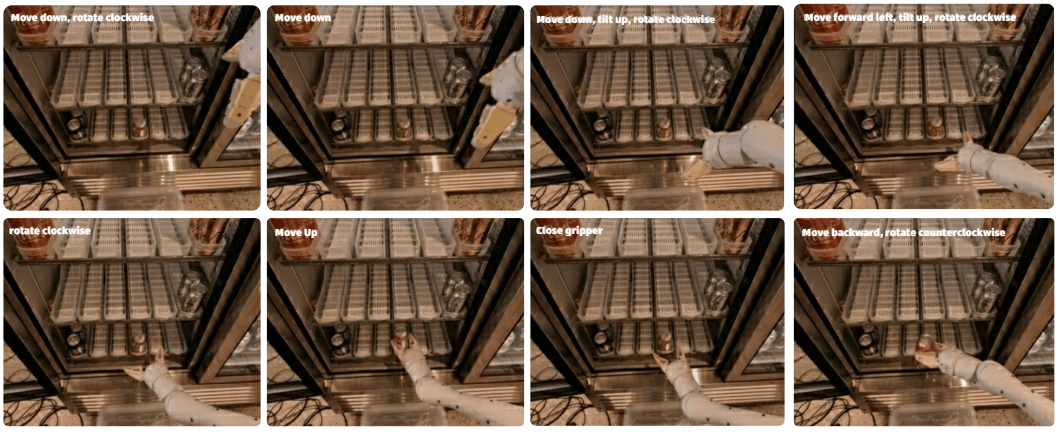



Figure 25: Language motion examples


Caption:
A square table with a white bowl containing an apple and two red fruits.

CR:
Question: Grab the apple from the serving container.
Answer: I will pick up the apple from the bowl.

Question: Could you take the round fruit from the white dish?
Answer: I will grab the apple from the dish.

CC:
Question: I need you to prepare the apple for cutting, please remove it from that dish.
Answer: I will take the apple out of the bowl.

Question: Can you put the apple on the counter so we can slice it for the salad?
Answer: I will take the apple out of the bowl.

QA:
Question: How many fruits are in the bowl?
Answer: Three.

Question: What is the color of the bowl on the table?
Answer: White.

Question: What is the shape of the table?
Answer: Square.

Ground Truth Instruction ✓

Caption:
A table with a bowl containing a bread roll and tomatoes.

CR:
Question: Move the bread roll out of the bowl.
Answer: Sure, I will lift the bread roll out.

Question: Remove the beige roll from the dish.
Answer: Alright, I will pick up the beige roll.

Question: Take the round brown bread out of the bowl.
Answer: Okay, I'll remove the round brown bread.

CC:
Question: I need the bread roll for a sandwich. Can you retrieve it from the bowl?
Answer: I will lift the bread roll out of the bowl.

Question: Please take the bread roll out of the bowl so we can start dinner.
Answer: I will take the bread roll out of the bowl.

Question: Could you remove the bread roll from the bowl in order to clean the table?
Answer: Sure, I will remove the bread roll.

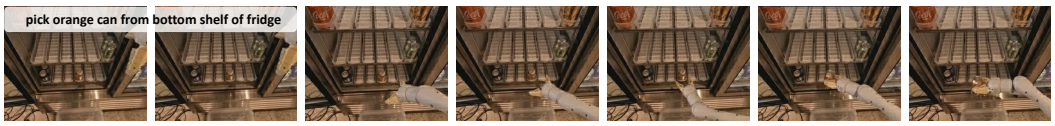
QA:
Question: What is inside the bowl?
Answer: A bread roll and tomatoes.

Question: Where can I find the bread roll?
Answer: In the bowl on the table.

Question: What is the color of the bread roll?
Answer: It is beige.

Ground Truth Instruction ✗

Figure 26: **Comparison of GPT annotations with and without ground truth instruction.** Errors are highlighted in red.




Caption:
A fridge with several empty trays and a few cans stored on the shelves.

CR:
Question: Retrieve the tangerine-colored can from the lower rack.
Answer: Okay, I will pick up the orange can from the bottom shelf.

Question: Remove the copper-hued can from the fridge's lowest compartment.
Answer: I will take out the orange can from the bottom shelf.

CC:
Question: I need you to collect the orange can from the fridge for the party.
Answer: I will pick the orange can from the bottom shelf.

Question: Get the beverage from the fridge so we can serve it to the guests.
Answer: I'll grab the orange can from the bottom shelf.

QA:
Question: How many cans are located on the bottom shelf?
Answer: Two cans are on the bottom shelf.

Question: What is the unique color can on the bottom shelf?
Answer: Orange.

Ground Truth Instruction ✓

Caption:
A vending machine with multiple trays containing various drink cans, including Coke, Sprite, and Pepsi.

CR:
Question: Grab the cylindrical silver container and place it in the empty tray below.
Answer: I'll pick up the silver can and place it in the tray below.

Question: Transfer the green can to the bottom shelf.
Answer: I will move the green can to the bottom shelf.

Question: Put the Pepsi can in the lowest compartment.
Answer: I'll put the Pepsi can in the bottom compartment.

CC:
Question: Organize the refrigerator by placing the Pepsi can from the third shelf to the bottom shelf.
Answer: I'll place the Pepsi can from the third to the bottom shelf to organize.

Question: Move the lemon-lime soda down to the last rack to make space for new items.
Answer: I will move the lemon-lime soda to the last rack.

Question: You're restocking the vending machine and need to place drinks in the bottom section. Begin with the Pepsi can.
Answer: I'll place the Pepsi can in the bottom section.

QA:
Question: How many Cola cans are in the refrigerator?
Answer: One.

Question: Which drinks are placed on the bottom shelf?
Answer: Pepsi and Sprite.

Question: Where is the Sprite can located?
Answer: On the third and bottom shelves.

Ground Truth Instruction ✗

Figure 27: **Comparison of GPT annotations with and without ground truth instruction.** Errors are highlighted in red. In this case, GPT-4o incorrectly infers the temporal sequence of actions without access to the instruction.

# E BENCHMARK DETAILS

## E.1 MULTIMODAL

We use the automatic evaluation from VLMEvalKit (Duan et al., 2024) including MMMU(Val) (Yue et al., 2024), MMStar (Chen et al., 2024a), MME (Fu et al., 2024), OCRBench (Liu et al., 2024e), HallB(Avg) (Guan et al., 2024), MMB(Dev En V1.1) (Liu et al., 2024d), TextVQA (Singh et al., 2019), DoCVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022) and RWQA (Team, 2024). These benchmarks collectively evaluate diverse multimodal capabilities, including general visual question answering, document, infographic and chart understanding, OCR reasoning, and hallucination robustness.

## E.2 SIMPLERENV-INSTRUCT

As shown in Table 14, although SimplerEnv-Instruct is primarily designed for instruction generalization, we incorporate diverse out-of-distribution objects, environments, and distractors to prevent VLA models from exploiting the benchmark by disregarding the instructions.

Table 14: Task distribution

| Attr. | with OOD Obj. | with OOD Env. | with Distract Obj. | Only Language OOD |
|---|---|---|---|---|
| Percentage(%) | 50.0 | 62.5 | 35.0 | 5.0 |

We select 10 task scenes with InstructVLA rollout actions and responses, as shown in Figures 28 and 29, to illustrate its performance on both task aggregation and Situated Reasoning tasks.

**Acknowledgements of 3D assets.** We gratefully acknowledge the creators of the following 3D assets used in *SimplerEnv-Instruct*. All assets are licensed under the Creative Commons Attribution license:

- *Fruit Bowl Collection*
  https://sketchfab.com/3d-models/fruit-bowl-collection-d89f6311cb0c4c37b0bf9cdd1e5abcb3
- *Coffee Mug*
  https://sketchfab.com/3d-models/coffee-mug-school-project-5f5ccee1514c440887c072fae8e0d699
- *Copper Tea Pot*
  https://sketchfab.com/3d-models/copper-tea-pot-27f2ac58f7614f2796630bdc6f18ee2f
- *Game Boy Classic*
  https://sketchfab.com/3d-models/game-boy-classic-0ae80019e6f046168923286d7e628f6f

All other assets are created using Blender or modified from SimplerEnv (Li et al., 2024d).

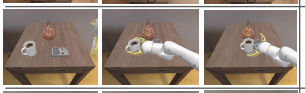
| Rollout Video | Instruction | Model Response |
|---|---|---|
|  | Please retrieve the green teardrop-shaped fruit from the surface.<br>Veuillez récupérer le fruit en forme de goutte verte de la surface. | I will pick up the pear. |
|  | "Pick up the yellow fruit on the table<br>saisis la banane | I will pick up the banana. |
|  | Pick up the small round purple fruit | I will pick up the grape. |
|  | Grab the green and white fizzy drink can from the workspace.<br>toma la lata verde y blanca | I will pick up the 7up can. |
|  | Please move the 'L' to 'V' | I will move the 'L' next to the 'V'. |
|  | pick the plastic bottle and then place it near the orange. | I will pick up the bottle and move it to the orange. |
|  | Can you place the sourest fruit to the less sour one? | I will put the lemon next to the pear. |
|  | pick the mug and then place it near the play station. | I will pick the mug and move it closer to the play station. |
|  | Pull the center drawer open | I will open the middle drawer now. |
|  | Pull the top drawer out<br>place the apple into the drawer you just opened | I will open the top drawer now.<br>I will open the top drawer and place the apple inside. |

Figure 28: **Examples of task aggregation in SimplerEnv-Instruct.** We list ten examples with corresponding instructions and responses. Notably, InstructVLA shows the strong **zero-shot** ability to interpret multilingual instructions, recognize novel objects, and leverage OCR capabilities.

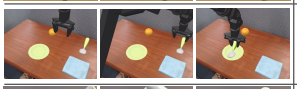
| Rollout Video | Instruction | Model Response |
|---|---|---|
|  | Please move the water bottle to wet the sponge. | I will pick up the water bottle to help you. |
|  | Please check if there are any items in the bottom drawer | The bottom drawer is currently closed, so I will open it to check for items.<br>The bottom drawer is already open, so I will check its contents. |
|  | I want a health vegetable. Grab it for me. | I will pick up the eggplant. |
|  | Can you grab the fruit before preparing the salad? | I will pick up the orange first. |
|  | I want to clean the table. Pick a suitable tool for me. | I will select the sponge for cleaning. |
|  | Open the top drawer and place coke can into it. | I will pick up the bottle and move it to the orange. |
|  | Pick the mug from the top drawer, then close the top drawer | The coffee mug is already on the table, so I will now close the top drawer. |
|  | put the tool that can be used to feed a baby on the plate | I will place the spoon on the plate. |
|  | After I finish my drink, can you place the fruit near it for a snack? | Sure, I will place the apple near the Pepsi can. |
|  | I am thirsty but I do not want drinks. Please grab something for me. | I will pick up the orange from the table. |

Figure 29: **Examples of Situated Reasoning in SimplerEnv-Instruct.** The second example's responses is recorded before and after the drawer is open.

# F Model Design and Training Details

## F.1 Instruction Format

To train captioning, question answering, and instruction-following capabilities, we integrate all tasks into a unified dialogue format. For captioning and question answering, we adopt the template shown in Prompt 3, where the captioning instruction is sampled from Prompt 4. For free-form instructions, we append the postfix "First answer my question." to elicit a direct response from the model, as illustrated in Prompt 5.

Dialogue Format

```
[
    {
        "role": "system", "content": DEFAULT_SYSTEM_MESSAGE
    },
    {
        "role": "user",
        "content": "[Question]",
        "image": image
    },
    {
        "role": "assistant",
        "content": "[Answer]"
    },
    {
        "role": "user",
        "content": "What action should the robot take to [Instruction]?"
    },
    {
        "role": "assistant",
        "content": "[Latent Action Queries]"
    }
]
```

Caption Prompts

- Describe what's on the table. Don't mention the robot arm.
- What objects are in the scene? Ignore the robot arm.
- Tell me what you see on the table, not the robot.
- Describe the items and their positions, but skip the robot.
- Look at the table and describe it. Don't include the arm.
- Only talk about the objects, not the machine.
- Give a short description of the scene, without the robot.
- Describe the setup on the table. Leave out the robotic arm.
- Focus on the objects and environment. Ignore the robot.
- Describe the environment and tabletop contents, excluding any robotic hardware.

Instruction Format

```
[
    {
        "role": "system", "content": DEFAULT_SYSTEM_MESSAGE
    },
    {
        "role": "user",
        "content": "What action should the robot take to [Instruction]? First answer my question.",
        "image": image
    },
    {
        "role": "assistant",
        "content": "[Response] [Latent Action Queries]"
    }
]
```

Table 15: **Model parameters.** "Adaptor" and "Scale Head" are used for MoE adaptation. Specifically, two LoRA adaptors are used to learn latent action generation and assistant response during VLA-IT.

| Component | Parameter | Value |
|---|---|---|
| Adaptor | Rank | 128 |
| | Alpha | 256 |
| | Dropout | 0.05 |
| | Target | Attn. Q/K/V/O |
| | | MLP Up/Down |
| Scale Head | Depth | 4 |
| | Size | 128 |
| Action Backbone | Depth | 12 |
| | Head | 12 |
| | Hidden Size | 768 |
| | RoPE Theta | 1000 |
| Proprioception Encoder(Optional) | Hidden Size | $8 \rightarrow 768 \rightarrow 768$ |
| | Activation | SiLU |
| Action Encoder with Time Embedding | Hidden Size | $7{+}768 \rightarrow 1536 \rightarrow 768$ |
| | Activation | SiLU |

Table 16: **Flow matching parameters.** The time steps is sampled from $p(\tau) = \beta(\frac{s-\tau}{s}; 1.5, 1)$ (Black et al., 2024)

| Component | Parameter | Value |
|---|---|---|
| Flow Sampling | s | 0.999 |
| | Inference Steps | 10 |
| Sinusoidal Time Embed | Max Period | 100 |

### F.2 Learning Objective and Inference Procedure

We adopt flow matching (Black et al., 2024; Lipman et al., 2022) to learn the action chunk $\mathbf{A} \in \mathbb{R}^{H\times 7}$ (Zhao et al., 2023) over a horizon $H$. The training objective is defined as the flow matching loss:

$$\mathcal{L}_{FM} = \mathbb{E}\left[\|V\theta(\mathbf{A}^\tau, q_t) - (\epsilon - \mathbf{A})\|^2\right], \tag{1}$$

where $\tau \in [0, 1)$ denotes the flow step, and $V_\theta(\mathbf{A}^\tau, q_t)$ is the network output conditioned on $q_t$, which encodes information from DINOv2 (Oquab et al., 2023) and a latent action $C$. The interpolated noisy action is given by $\mathbf{A}^\tau = \tau\mathbf{A} + (1-\tau)\epsilon$, with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

During inference, we generate the action chunk using forward Euler integration:

$$\mathbf{A}^{\tau+1/N} = \mathbf{A}^\tau + \frac{1}{N} V_\theta(\mathbf{A}^\tau, q_t), \tag{2}$$

starting from $\mathbf{A}^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with $N = 10$ denoising steps.

### F.3 Model Parameters

Additional model parameters are provided in Table 15, with flow-matching sampling settings detailed in Table 16. All projectors—including those aligning latent actions and DINO-ViT visual features to the action expert's dimension—use a simple two-layer MLP with SiLU activation. The action head, also a shallow MLP with SiLU, maps the action expert's hidden states to $\mathbb{R}^{N\times 7}$, where $N = 16$ is the prediction horizon and 7 denotes the action dimension, including the gripper.

### F.4 Inference Speed

We evaluate the inference speed of InstructVLA on a single A100 GPU with BF16 precision, as shown in Table 17. To support language feedback during evaluation (i.e., CoT inference), in the "Thinking" setting, we enable VLM auto-regressive generation every 20 action expert steps. The "Action Only" setting bypasses language generation and directly decodes latent actions via a single VLM forward pass. In the "Latent Action Caching", latent actions are generated every two expert steps; this introduces minimal performance impact. All settings are tested without action chunking. Note that although the model predicts 16-step action sequences, only one step is executed.

Table 17: **Inference speed.** Inference speed is evaluated under three settings **without using action chunking**. Each evaluation includes a 50-step warm-up followed by 200 steps for stable measurement.

| | With Language | Action Only | Latent Action Caching |
|---|---|---|---|
| Inference Frequency(Hz) | 2.51 | 3.50 | 4.96 |

### F.5 Experiments Compute Resources

The action pretraining phase requires approximately 27 hours on 64 A100 GPUs, with each node equipped with 1 TB of CPU memory. The VLA-IT phase takes about 12 hours under the same GPU configuration. Simulator-based evaluations are conducted with 8 A100 GPUs, while real-world experiments involve 4 hours of training on 32 A100 GPUs and deployment on a single A100 GPU.

To assess minimal training resources, we further reproduce pretraining results using 8 A800 GPUs in 2.5 days as shown in Table 18.

Table 18: **Evaluation results under different training settings.** We report mean success rates (%± standard error) across tasks, with Overall denoting the average over all tasks. "Main" corresponds to the results reported in the main table.

| | Google Robot (VA) | | | | Google Robot (VM) | | | | WidowX Robot | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Setting | Pick Coke | Move Near | Drawer | Apple In | Pick Coke | Move Near | Drawer | Apple In | Put Spoon | Put Carrot | Stack Cube | Put Eggplant | |
| Main (8 epochs) | 92.3±0.7 | 71.9±1.3 | 61.7±0.8 | 33.1±2.5 | 79.6±1.9 | 68.3±3.1 | 52.3±3.8 | 50.3±3.8 | 43.1±6.4 | 40.3±14.6 | 9.7±9.6 | 94.4±2.4 | 56.2±2.9 |
| 8 GPUs (4 epochs) | 94.0±0.2 | 76.9±0.5 | 62.8±1.6 | 39.3±4.3 | 88.7±1.7 | 67.4±2.1 | 61.8±2.5 | 31.7±1.9 | 62.5±11.0 | 48.6±2.4 | 8.3±4.2 | 95.8±4.1 | 61.5±1.3 |

## G MULTIMODAL EXAMPLES

Figure 30 illustrates InstructVLA's multimodal and embodied commonsense reasoning across diverse scenarios. The model demonstrates accurate visual inference (e.g., recognizing a dog via reflection, identifying synthetic images), basic scene text recognition, and reliable grounding of objects and colors. In manipulation tasks, it interprets high-level goals, predicts appropriate next actions, and verifies task completion. These capabilities showcase its integration of perception, language, and manipulation, enabling effective performance in complex daily-life scenarios.

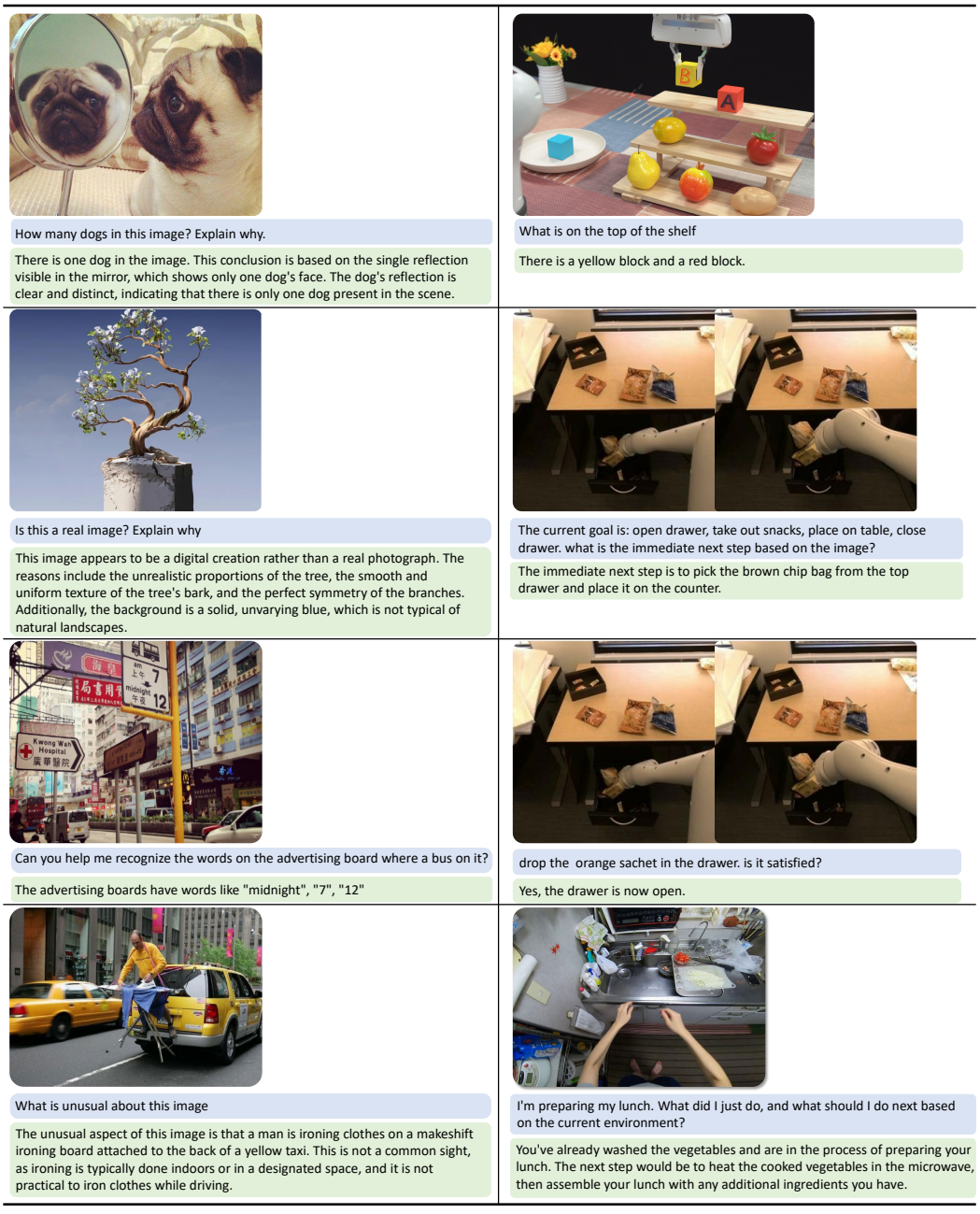



Figure 30: **Zero-shot multimodal question answering.** Four commonsense and four embodied examples are selected.

## H Real-world Experiments Setup and Analysis

We collect data exclusively for few-shot settings as shown in Figure 31. In the first setting, which focuses on grasping objects in a clustered arrangement, the robot is instructed to classify objects within a $20 \times 40$ cm region on the table—placing all cubic objects into a plate and all others into a box. This setting includes 70 complete episodes, totaling 677 pick-and-place actions. In the second setting, which emphasizes spatial actions, the robot is instructed to randomly grasp three objects from the top of a rack and place them into a plate. We collect 60 complete episodes for this setting, comprising 180 pick-and-place actions. The experimental setups are depicted in Figure 35.

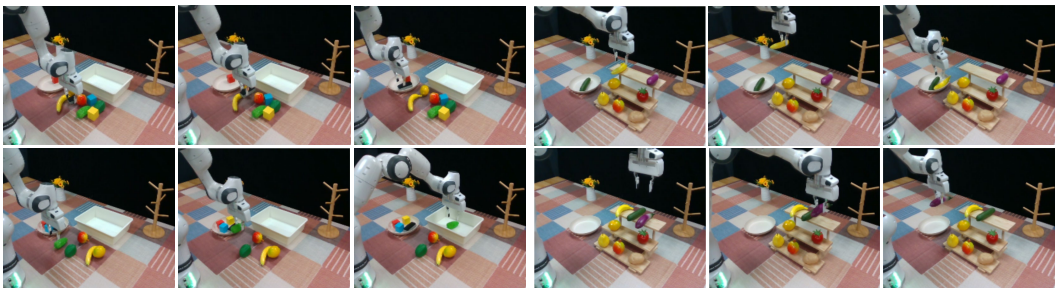


Figure 31: **Real-world dataset examples.** Four examples from the few-shot training set, illustrating cluster classification tasks (left) and rack pick-and-place tasks (right).

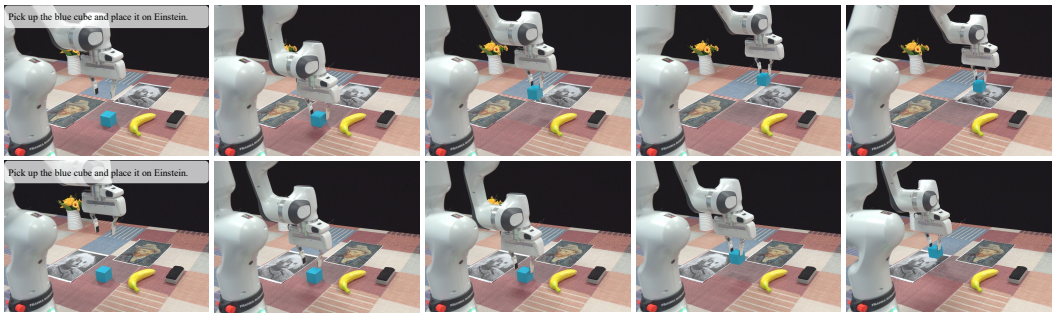



Figure 32: **Zero-shot grounding.** In a clustered pick-and-place setting, InstructVLA accurately places the blue cube by semantically grounding the reference to the celebrity.

Figure 33: **Light distraction.** Stable visual features from DINO and SigLIP enable the model to operate robustly under extreme out-of-distribution lighting conditions.

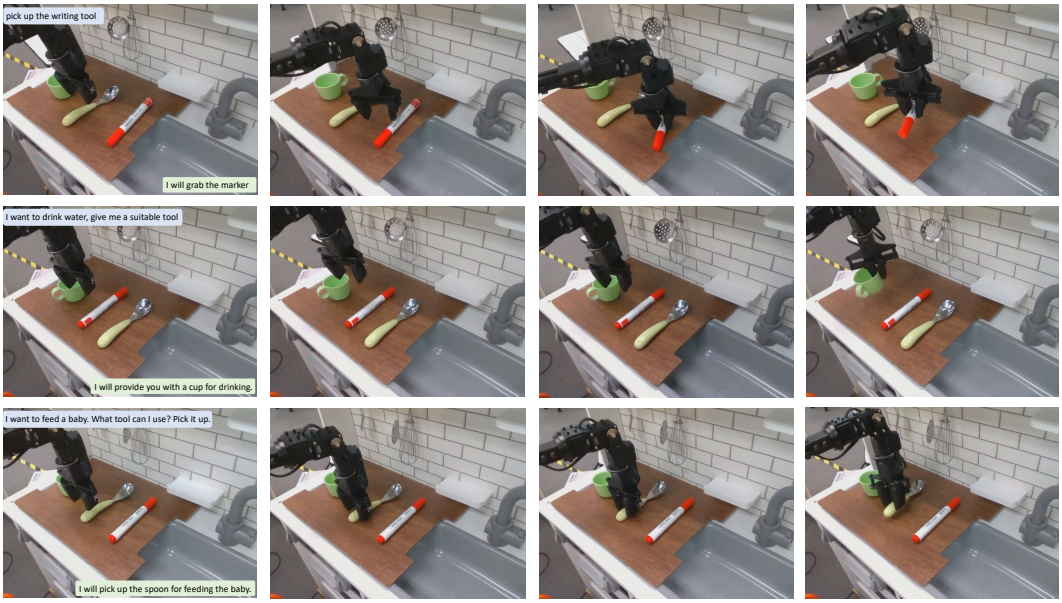



Figure 34: **Zero-shot evaluation.** We perform zero-shot evaluation in the Bridge kitchen environment with augmented background and novel objects. The instruction and model response are visualized in the first image.

To assess semantic grounding in novel contexts, we replace the plate and box in the cluster classification setting with images of celebrities. As illustrated in Figure 32, the model accurately interprets instructions and places the blue cube correctly by leveraging object and celebrity recognition.

Figure 33 shows that InstructVLA remains robust under extreme lighting conditions, supported by stable visual features from DINO and SigLIP. Finally, we evaluate zero-shot generalization in the Bridge kitchen environment with augmented backgrounds and unfamiliar objects. As shown in Figure 34, the model successfully follows novel instructions and completes the tasks.

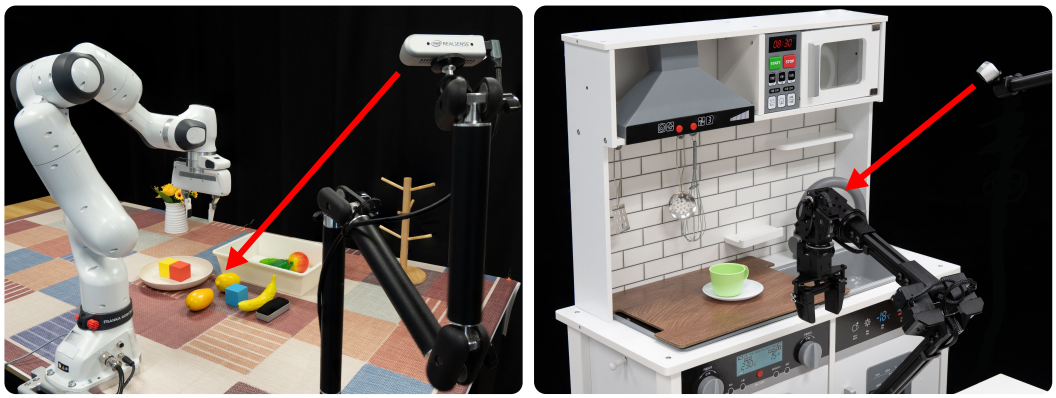


Figure 35: **Real-world settings.** A third-person view is captured using an Intel D435i camera for the Franka (few-shot) and WidowX (zero-shot) settings.

# I Broader Impacts and Future Work

## I.1 Limitation

InstructVLA integrates world knowledge into manipulation tasks by performing multimodal reasoning prior to action generation. Recent VLMs also excel at long-context processing and multi-turn dialogue. This motivates curating interleaved manipulation and reasoning with multi-turn interaction to support long-horizon tasks involving user intervention or reasoning-action alternation (Yao et al., 2023). Furthermore, existing tasks remain limited to basic primitives such as open/close and pick/place by the datasets we use (Brohan et al., 2022; Ebert et al., 2021). Extending InstructVLA to more dexterous skills is essential for real-world deployment.

## I.2 LLM Usage Statement

We employed large language models (LLMs) solely for grammar refinement and minor linguistic polishing. All LLM-assisted edits were carefully reviewed and verified by the authors to ensure that no fabricated content or unintended alterations to the original meaning were introduced. The research ideas, experimental design, data analysis, and conclusions presented in this work were entirely conceived and executed by the authors without LLM assistance.

## I.3 Broader Impacts

InstructVLA contributes to the advancement of general-purpose embodied agents by integrating vision-language understanding with action generation. Its ability to follow free-form instructions and generalize to novel tasks supports applications in assistive robotics and human-robot collaboration. Nonetheless, as with other large pretrained models, careful attention must be given to potential limitations such as dataset bias and safety in real-world deployment. Ensuring responsible use and reliable performance across diverse environments is essential.

## I.4 Future Work

We plan to incorporate additional sensory modalities, such as depth and tactile feedback, to enhance safety and reliability in physical interactions. Leveraging recent advances in digital twins and simulation technologies, we aim to reduce reliance on real-world data by utilizing large-scale synthetic datasets. Finally, we will extend the evaluation and deployment of InstructVLA to a broader range of environments to further assess its generalization capabilities.