From Grounding to Manipulation: Case Studies of Foundation Model Integration in Embodied Robotic Systems

Anonymous ACL submission

Abstract

Foundation models (FMs) are increasingly used to bridge language and action in embodied agents, yet the operational characteristics of different FM integration strategies remain underexplored—particularly for complex instruction following and versatile action generation in changing environments. This paper examines three paradigms for building robotic systems: end-to-end vision-language-action (VLA) models that implicitly integrate perception and planning, and modular pipelines incorporating either vision-language models (VLMs) or multimodal large language models (LLMs). We evaluate these paradigms through two focused case studies: a complex instruction grounding task assessing fine-grained instruction understanding and cross-modal disambiguation, and an object manipulation task targeting skill transfer via VLA finetuning. Our experiments in zero-shot and few-shot settings reveal trade-offs in generalization and data efficiency. By exploring performance limits, we distill design implications for developing language-driven physical agents and outline emerging challenges and opportunities for FM-powered robotics in real-world conditions.

1 Introduction

Natural language is emerging as a universal interface for embodied robotic systems. Recent foundation models (FMs) allow robots to follow free-form instructions in perception, reasoning, and motor commands, offering the promise of *language-grounded autonomy*. They include multimodal large language models (LLMs) (Grattafiori et al., 2024; Bai et al., 2025; Lu et al., 2024), vision–language models (VLMs) (Liu et al., 2024a; Ravi et al., 2025; Ren et al., 2024; Li et al., 2023), and vision–language–action (VLA) models (Kim et al., 2024; Zheng et al., 2025; Qu et al., 2025; Bu et al., 2025).

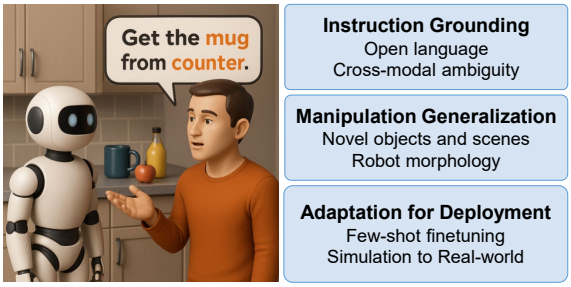

Figure 1: Challenges of foundation models in embodied robotic systems include cross-modal instruction grounding, generalization across environments and morphologies, and data-efficient adaptation for the real world.

However, turning the promise of *language-grounded autonomy* into deployable systems is highly challenging. Robots must (i) map ambiguous instructions to the physical world (*instruction grounding*), (ii) execute reliably across novel objects, scenes, and robot morphologies (*generalizable execution*), and (iii) achieving the aforementioned goals with limited data (*efficient adaptation*). How well different FM integration strategies meet these competing requirements remains underexplored (Fig. 1).

To our best knowledge, this work delivers the first head-to-head empirical comparison of three prevalent integration paradigms: end-to-end VLAs that directly map language and vision to actions, multimodal LLM agents that orchestrate perception and control through tool calls, and modular VLM pipelines that couple perception-specialist FMs with task-specific planners (Fig. 2; Table 1).

We evaluate these paradigms through tabletop case studies designed to highlight their complementary strengths and limitations. Specifically, we consider two task categories: (i) Complex Instruction Grounding, which probes fine-grained understanding and cross-modal disambiguation (Sec.3); and (ii) Object Manipulation, which measures the ability to transfer learned skills after VLA fine-tuning

under distribution shifts, complemented by comparative and ablation studies (Sec.4).

Our zero-shot grounding experiments reveal distinct trade-offs across integration strategies. VLM pipelines prioritize interpretability and data efficiency, sacrificing flexibility and peak performance. While underperforming on handling complex instruction grounding, they deliver moderate performance in object grounding—using less than 1% of the parameters required by multimodal LLMs. In contrast, multimodal LLM agents generalize better on complex instructions but incur significantly higher inference costs. Notably, smaller reasoning-focused models such as GPT-4o-mini can even outperform larger models like GPT-4o on certain tasks. VLAs, with their tightly coupled perception-to-action pathways, support streamlined action generation, yet struggle to reason about rare or abstract concepts. We further examine the trade-off between model size and performance by analyzing quantization effects on open-source multimodal LLMs. These findings offer practical guidance for developing language-driven robotic systems under real-world constraints.

Within the VLA paradigm, we categorize models by their action generation mechanisms—autoregressive (Vaswani et al., 2017; Kim et al., 2024; Hung et al., 2025) and diffusion-based approaches (Ho et al., 2020; Chi et al., 2023; Reuss et al., 2024; Wen et al., 2024). We evaluate their adaptability through fine-tuning under distribution shifts, mirroring real-world deployment scenarios. To assess generalization, we analyze robustness to environmental perturbations and variation in object appearance and robot morphology.

To summarize, the main contributions of this work are as follows:

- To our best knowledge, we present the first systematic comparison of *end-to-end VLA*, *modular VLM*, and *modular multimodal LLM* architectures on a same set of embodied tasks.
- We release a dataset and accompanying code that supports evaluation of complex instruction grounding and manipulation transfer—covering accordance comprehension, cross-modal reasoning, and motor adaptation.
- We benchmark state-of-the-art VLAs and multimodal LLMs, offering timely insight into the capabilities and failure modes of modern FMs in robotics.
- We distill actionable trade-offs that practitioners can apply when choosing an FM stack for language-driven embodied agents.
- We release all code, data to facilitate reproducible embodied AI studies; We also release a complete, end-to-end claw-machine robot system to demonstrate FM integrations in real-world applications.

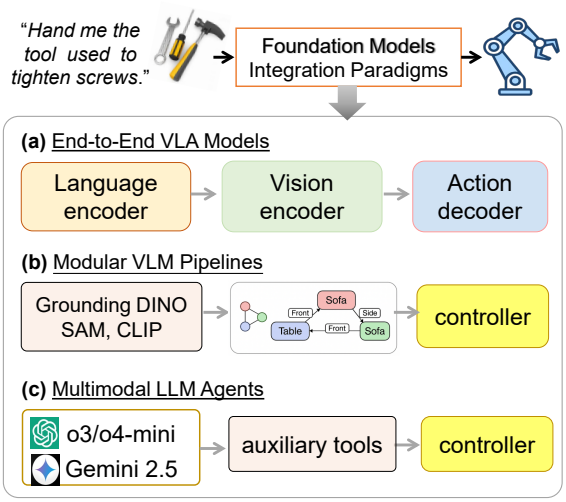

Figure 2: Foundation model integration strategies in language driven robots: (a) End-to-end VLA models, (b) Modular VLM pipelines, and (c) Multimodal LLM agents. Each strategy reflects a distinct interface between language, perception, and control.

2 Foundation Model Integration for Language-Guided Robotics

Concerning how FMs are integrated into robot systems, we identified the following three types of integration strategies (Fig. 2). In the following, We briefly describe each strategy along with its respective advantages and limitations.

2.1 End-to-End Vision-Language-Action

Definition. Vision-Language-Action (VLA) models operate in an end-to-end manner, directly translating visual observations and natural language instructions into low-level actions without decoupled perception, language, and control modules (Fig. 2a). Two mainstream paradigms have emerged within this framework: auto-regressive and diffusion-based action generation. Through large-scale pretraining, these models acquire broad capabilities that support generalization across tasks. However, efficient adaptation to real-world settings remains a significant challenge.

Pipelines for Robot Systems	Instruction Grounding			Manipulation Generalization		Adaptation for Deployment
	Visual inputs	Multi-round dialogue	CoT reasoning	Morphology independent	Skill sets	Data Efficiency
End-to-End VLA Models	✓	✗	✗	✗	Wide range	Data-hungry finetuning
Modular VLM pipelines	✓	✗	✗	✓	Controller-specific	Cheap finetuning
Multimodal LLMs Agents	✓	✓	✓	✓		In-context learning

Table 1: Comparison of foundation model integration strategies in embodied robotic systems, highlighting differences in instruction grounding, manipulation generalization, and adaptation methods.

Autoregressive VLA Models. Autoregressive VLA models typically process language and visual inputs, employing various tokenization strategies to convert multimodal data into a unified latent space. Then the transformer-based decoder generates actions step-by-step in an autoregressive manner conditioned on the input context and previously generated actions, allowing structured action generation and planning.

Building on the previous groundbreaking models such as RT-1 (Brohan et al., 2023), RT-2 (Zitkovich et al., 2023) and VIMA (Jiang et al., 2023), OpenVLA (Kim et al., 2024) emerges as an important open-source method, combining a fine-tuned Llama 2 (7B) model with DinoV2 and SigLIP for visual tokenization, pretrained on the Open-X-Embodiment dataset (O'Neill et al., 2024) consisting of 970k real-world robot demonstrations. TraceVLA (Zheng et al., 2025) improves OpenVLA with visual trace prompting to enhance spatiotemporal awareness. Emma-X (Sun et al., 2024) further refines dataset quality using a trajectory segmentation and Embodied Chain-of-Thought reasoning. This work is followed by NORA (Hung et al., 2025), which uses Qwen-2.5-VL-3B as backbone and pretrained on the Open-X-Embodiment dataset. In parallel, other efforts integrate tactile sensing into robot perception (Yang et al., 2024; Gao et al., 2024; Zhao et al., 2024), supporting more generic robot policies. Other recent developments include distilling spatial representations from VLMs, e.g., Gemini-Robitics (Gemini Robotics Team, 2025), and enriching training with additional features, such as 3D spatial relationships in SpatialVLA (Qu et al., 2025), task-centric latent space in UniVLA (Bu et al., 2025), and integrating multimodal understanding with action prediction in ChatVLA (Zhou et al., 2025) and UP-VLA (Zhang et al., 2025).

Diffusion-based VLA Models. Diffusion-based VLA models formulate action generation as a denoising process over latent trajectories. Given a noisy version of a full action sequence, the model learns to recover the true trajectory conditioned on language and vision.

Diffusion Policy (DP) (Chi et al., 2023) pioneered the use of diffusion model for visuomotor policy representation, laying the foundation for subsequent multimodal approaches. Octo (Octo Model Team et al., 2024) uses conditional diffusion decoding for action sequence prediction. Rather than lightweight diffusion heads, (Reuss et al., 2024; Wen et al., 2024; Li et al., 2024b) use larger and more dedicated diffusion policy modules the as action decoder. DTP (Fan et al., 2025) introduces a trajectory-level guidance to enhance diffusion-based planning. Recent works such as π_0 (Black et al., 2024) and $\pi_{0.5}$ (Intelligence et al., 2025) integrate a pretrained VLM with a flow-matching-based action expert to model action distributions. These models often utilize FAST (Pertsch et al., 2025) for efficient compressed action tokenization. HybridVLA (Liu et al., 2025) unifies diffusion and autoregressive action prediction within a single LLM-based framework. A growing focus is placed on adapting VLA models to diverse embodiments including bimanual manipulation and humanoid robots, such as RDT-1b (Liu et al., 2024b), DexVLA (Wen et al., 2025) and GR00T N1 (Bjorck et al., 2025).

Strengths and Limitations. VLA models provide a unified, end-to-end framework for robotic manipulation, (i) seamlessly integrating visual, language, and action modalities. Leveraging large-scale pretraining and fine-tuning, these models exhibit (ii) strong potential for generalizing across diverse manipulation tasks and robotic embodiments. However, their performance is constrained by the (iii) *limited availability* of high-quality, diverse robotic datasets. Pretraining can also introduce biases from training distributions, leading to (iv) degraded performance in out-of-distribution scenarios, such as *novel tasks* or *different robotic*

embodiments. Therefore, despite their potential, further work is needed to enhance their robustness and generalization across real-world settings, for example, efficient adaptation using few-shot data.

2.2 Modular Vision–Language Pipelines

Definition. In a modular vision-language pipeline (Fig. 2b), perception is handled by a *specialist* vision language model (VLM) that outputs symbolic scene information, typically grounded 2-D / 3-D bounding boxes, segmentation masks, or referring expression pointers. A downstream planner or policy module then consumes this structured representation to generate low-level actions. The language channel is therefore *disentangled* from motor control, allowing each module to be tuned independently, thus preserving the transparency and plug-and-play advantages of classical planning.

Representative systems. Language-promptable specialist VLMs endow modular stacks with zero-shot semantics for various robotics pipelines. (Bandyopadhyay et al., 2024) demonstrates an end-to-end *sample collection robot system* that uses GroundingDINO (Liu et al., 2024a) to localize objects and refines each box with SAM (Ravi et al., 2025) masks before passing them to classical grasp-and-place controllers, illustrating this paradigm's practicality in real deployments. (Werby et al., 2024) aggregates these modules into a floor–room–object hierarchy, showcasing their usage in long-horizon language-conditioned navigation across multi-story buildings.

Strength and Limitations. Modular VLM pipelines strike a balance between transparency and adaptability, and delivers practical benefits: (i) *interpretability*—detections can be inspected; (ii) *lightweight*—the model parameters are usually around 100M∼600M, approximately 1% $\sim$ 6% the size of LLaMA 3.2 Vision 11B (Grattafiori et al., 2024). On the other hand, it is limited at (i) *interaction rigidness* compared with more flexible multimodal LLMs, and (ii) *pipeline brittleness* where perception errors propagate without mitigation (Fig. 2b; Table 1). Their success hinges on robust open-vocabulary grounding—precisely the capability our *Instruction Grounding* case study stresses in Section 3.

2.3 Multimodal LLM Agents as Orchestrators

Definition. Multimodal LLM agents place a large, tool-calling language model at the centre of the control loop (Fig. 2c). The LLM receives raw user utterances, selectively invokes vision tools (*e.g.*, a detector or depth estimator) via function calls, reasons over their outputs in-context, and finally issues high-level action primitives to a low-level controller. The agent therefore acts as a *cognitive hub* that binds perception and control through natural language.

Representative Systems. Multimodal LLMs are taking increasingly important roles in robotics. Gemini Robotics (Gemini Robotics Team, 2025) integrates perception, spatial reasoning, and trajectory synthesis into one Gemini-2.0 backbone (Google DeepMind, 2024), which serves as the embodied brain. (Li et al., 2024c), in the same vibe as Gemini Robotics, leverages the inherent common sense and reasoning capabilities of these models by fine-tuning adapter modules through a chain-of-thought training paradigm. It endows the model with accurate pose prediction and precise manipulation abilities. These works collectively show the trend that the multimodal LLM shifts to the "cognitive hub" in robot systems. (Glocker et al., 2025) build a modular agent-orchestration system for household object management robots. It utilize Llama 3.2 Vision (Grattafiori et al., 2024) for open-vocabulary perception to facilitate creating grounded task plans, while the limitations of the multmodal LLM were not discussed.

Somewhat similar to our work, (Li et al., 2024a) investigates the eligibility of Multimodal LLMs to serve as the "brain" for in-home robotics by providing a benchmark to compare models along the axes of perception, visual reasoning and task planing. Models like GPT-4V, Qwen-VL (Bai et al., 2025) and DeepSeek-VL (Lu et al., 2024) were included, but more recent releases were not covered—likely due to the fact that the field is moving fast with new models emerging in rapid succession.

Strengths and Limitations. Multimodal LLM agents excel in (i) *visual commonsense reasoning*, leveraging extensive language priors to generalize to novel concepts beyond the reach of most specialist VLMs, and (ii) *instruction following* with support for fine-grained visual understanding and dynamic planning. Despite their expressive power, however, these models are (iii) *resource-intensive*,

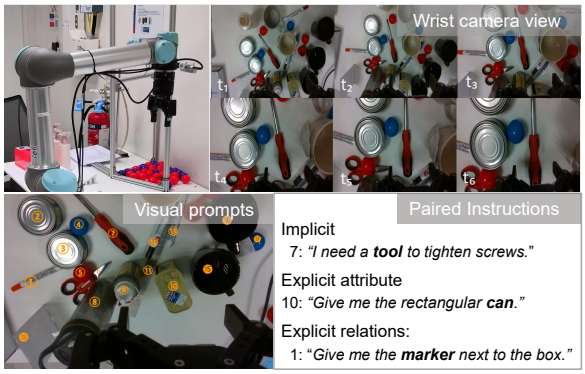

Figure 3: Experimental setup for two case studies in a cluttered tabletop environment. The top row shows egocentric video data collected for the manipulation case study. The bottom row is an example setup for the instruction grounding task, including an annotated visual prompt paired with complex instructions in three forms: implicit, explicit with attributes and spatial references.

MODEL	EASY	MEDIUM	HARD	AVG
Specialist VLMs				
GroundingDINO-86M	0.518	0.357	0.349	0.408
GroundingDINO-145M	0.443	0.320	0.355	0.372
Closed-source Multimodal LLMs				
Gemini2.5-Pro-Exp	**0.904**	**0.765**	**0.793**	**0.821**
Gemini2.0-Flash	0.884	0.738	0.678	0.767
GPT-4.5	0.837	0.723	0.739	0.766
GPT-4o	0.814	0.745	0.683	0.747
GPT-4o-mini	0.803	0.722	0.604	0.710
o4-mini	0.721	0.769	0.710	0.733
GPT-4V	0.470	0.476	0.467	0.471
Open-source Multimodal LLMs				
Llama-3.2V-90B	0.722	0.701	0.657	0.693
Llama-3.2V-11B	0.583	0.569	0.547	0.566
Llama-4-Maverick	0.698	0.576	0.634	0.636
Llama-4-Scout	0.776	0.615	0.624	0.672
Qwen2-VL-72B	0.686	0.614	0.558	0.619
Gemma-3-27B	0.452	0.384	0.267	0.368
DS-Janus-Pro-7B	0.444	0.330	0.317	0.364
Phi-3.5-Vision-4.2B	0.291	0.357	0.205	0.284

Table 2: Object grounding performance of specialist VLMs and multimodal LLMs (closed-source and open-source) across varying scene complexity levels. Models are evaluated on easy, medium, and hard cluttered scenes, with macro accuracy reported.

posing challenges for deployment—particularly on mobile robotic platforms.

3 Case Studies on Instruction Grounding

Natural language *instruction grounding* involves translating user intents into clear, actionable goals in a visual scene, which is a key capability for embodied AI (Gemini Robotics Team, 2025). Our case study offers empirical insights into the grounding performance of various models through the lens of challenging cross-modal disambiguation, and further examines the trade-offs introduced by model sizes and quantization—providing practical suggestions for efficient deployment.

Benchmark Dataset. To minimize the impact of vision priors on measured performance, we design benchmarking scenarios using household objects placed on a tabletop. These objects are commonly represented in the training datasets of the foundation models, and the tabletop setup features minimal variation in lighting and camera angles—ensuring that the evaluation primarily reflects grounding capabilities.

We curated a new Instruction Grounding benchmark dataset. In images containing multiple household objects, each object is tagged with a number as the visual prompt, and each image is paired with language instructions crafted to test visual commonsense and cross-modal disambiguation concerning attribute or spatial relationships – for "*pick up the red-capped marker*," the color must be used to select one among a few markers; whereas "*grasp the cup in front of the screwdriver*" requires reasoning over spatial relations (Fig. 3; Appendix).

Language ambiguity often leads to execution failure in an embodied system. By comparing specialist VLMs and multimodal LLMs, we reveal their concrete failure modes that further inform our design implications in Sec. 5.

Zero-Shot Object Grounding. We begin with a foundational question for instruction grounding: *Can FMs accurately recognize objects in cluttered open scenes?* Table 2 presents the performance hierarchy for specialist VLMs and a range of multimodal LLMs, serving as a basis for deeper analysis of ambiguity resolution in later sections.

- Despite their popularity in modular pipelines, GroundingDINO achieve only 35–41% accuracy, as it struggles with featureless objects, e.g. *'the can'*. Moreover, it is brittle in open scenes, e.g. a *'screwdriver'* is constantly recognized as a *'marker'*, which instead is an easy case for multimodal LLMs which embodied large volume of visual commonsense.
- Gemini 2.5-Pro takes the first place with 0.82, followed by Gemini 2.0-Flash and GPT-4.5. Open-source systems still trail the proprietary tier. Llama 3.2-Vision 90B reaches about 84%

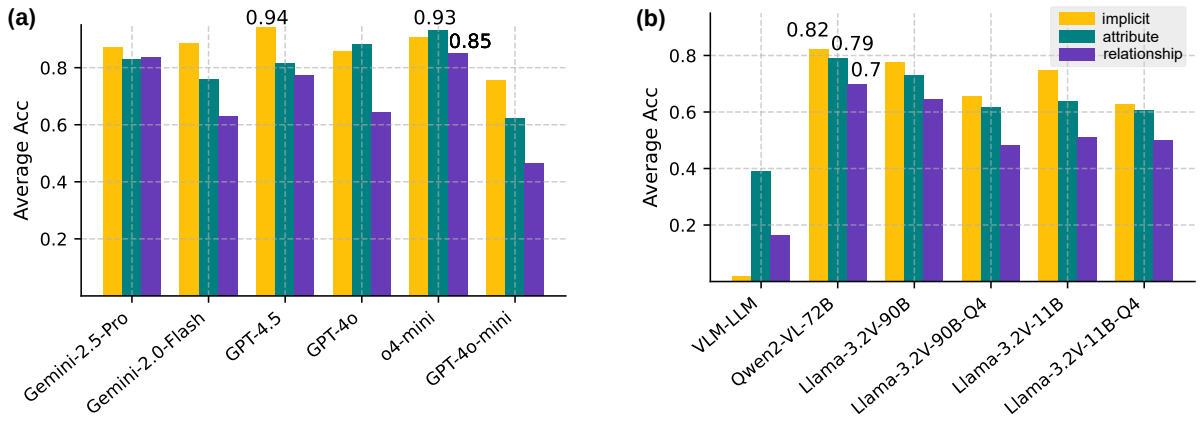

Figure 4: Performance of complex instruction grounding across VLM–LLM pipelines and end-to-end multimodal LLMs. Macro accuracy is reported across instruction types—implicit, attribute-based, and relationship-based. Subfigures show (a) proprietary models and (b) open-source models along with their Int4-quantized variants.

of Gemini 2.5's score, while the more recent Llama 4 releases did not outperform it.

- Smaller community models (Gemma-27B, Phi-Vision) fall below the specialist-threshold, suggesting that they are still inadequate for fine-grained grounding in cluttered scenarios.
- Last, when compute is a bottleneck, GPT-4o-mini (0.71) and Llama 3.2-Vision 11B (0.57) provide the best speed–accuracy trade-off, delivering decent performance without incurring heavy memory footprint or high API costs.

Zero-Shot Complex Instruction Grounding. This task is framed as a multiple-choice problem, where the model is asked to select the correct object index in a cluttered scene based on three types of natural language instructions: implicit, attribute-based, and relationship-based—each type probes a distinct grounding challenge. We evaluate a series of multimodal LLMs, using a modular VLM–LLM pipeline as a baseline. In this pipeline, the LLM parses the instruction to infer likely targets, queries GroundingDINO to detect candidate objects, and selects from the detected boxes—essentially guessing without directly perceiving the scene.

- *Implicit Instruction Grounding*. Instructions like "I need a tool to tighten the screws" only refer to the target object implicitly, and the model needs to infer the target object using its common sense priors. For such instructions, the modular VLM–LLM pipeline struggles to select a screwdriver, lacking embedded affordance reasoning. In contrast, multimodal LLMs perform well, reflecting strong visual commonsense. GPT-4.5 demonstrates exceptional performance (0.94), though its high inference cost—20× that of Gemini 2.5 makes it cost-prohibitive for most applications.
- *Relational Reasoning Remains Challenging.* This category requires resolving referential ambiguity through implicit chain-of-thought reasoning: grounding objects, modeling spatial relationships, and disambiguating targets (e.g., identifying the correct mug among many based on "next to something"). Accuracy drops significantly nearly across all models. Only Gemini 2.5-Pro and o4-mini achieve accuracy above 0.80—the former likely benefits from embodied training data, while the latter demonstrates strong reasoning capabilities. Notably, o4-mini is a medium-sized model, yet it outperforms larger models like GPT-4.5 on relational instructions—suggesting that structured reasoning may help close, or even overcome the performance gap brought by different model scales.
- *Instruction-Dependent Quantization Effects.* INT4 quantization reduces the model size by over 70%, making it an attractive choice for deployment. In Llama 3.2 Vision, we observe that it disproportionately impacts implicit and relational instruction grounding, indicated by the relative accuracy drop of $14\% - 17\%$, while attribute grounding is more robust with only 4% loss. Despite reduced precision, quantized 11B models offer a speed–accuracy balance for low-resource settings. Our findings underscore the need for *fine-grained quantization strategies* that preserve the most important high-level reasoning capabilities under resource constraints.

MODELS	LIBERO-SPATIAL	LIBERO-OBJECT	LIBERO-GOAL	LIBERO-LONG	AVERAGE
OpenVLA finetuned	84.7	88.4	79.2	53.7	76.5
π_0 finetuned	**96.8**	**98.8**	**95.8**	**85.2**	**94.15**
π_0-FAST finetuned	96.4	96.8	88.6	60.2	85.5
SpatialVLA finetuned-AC	88.2	89.9	78.6	55.5	78.1
NORA-finetuned	85.6	87.8	77.0	45.0	73.9
NORA-finetuned-AC	85.6	89.4	80.0	63.0	79.5
NORA-Long-finetuned	92.2	95.4	89.4	74.6	87.9

Table 3: Success rates (%) on the LIBERO Simulation Benchmark across four task suites, each evaluated over 500 trials. Results for SpatialVLA are from (Qu et al., 2025); Results for π_0 are from (Black et al., 2024), using pretrained models on LIBERO benchmarks. "AC" denotes the use of action chunking. The comparison in the Appendix highlights its impact on performance. The finetuned π_0 model achieves the highest performance.

4 Case Studies on Robotic Manipulation

Now we shift the focus to *skill adaptation*. In an ideal deployment scenario, a pretrained VLA—already endowed with broad visuomotor skills—should be retargeted to a new manipulation task with minimal data and fast convergence. We use fine-tuning, the standard practice for adaptation, as a probing lever to evaluate how the state-of-the-art VLA models adapt to new tasks and deployment conditions.

Given the scale of VLAs, we compare **partial fine-tuning**, which leverages our benchmark dataset (Appendix) and its inherent distribution bias to study convergence behavior, and **full fine-tuning**, which uses large-scale datasets to minimize the training loss. Our evaluation focuses on three key aspects: (i) *training dynamics*—how quickly and smoothly training converges; (ii) *generalization*—how well the resulting policies perform on various tasks; and (iii) *robustness*—how well the resulting policies handle environmental distractors. Our experiments highlight the performance of VLA models in different settings, offering practical suggestions for practitioners who have to adapt large VLAs under tight data, time and compute budgets.

Skill Adaptation Performance. Our fine-tuning process consists of two stages: (1) To assess convergence behavior under distribution shift, we collected a custom dataset (see Appendix for details) with a distribution bias relative to common pretraining datasets included in the Open-X-embodiment (O'Neill et al., 2024) and LIBERO datasets (Liu et al., 2023). We used it to partially fine-tune several recent VLA models and trained Diffusion Policy (DP) and Action Chunking Transformer (ACT) from scratch. The results are shown in Fig. 5; (2) For full fine-tuning, we leveraged larger benchmark datasets, Open-X-embodiment and LIBERO, to fully fine-tune RT-1, OpenVLA, SpatialVLA and NORA, and compared their performance. The results are shown in Fig. 6.

- *Partial Fine-tuning.* Through the experiments we observe that DP and ACT exhibit high stability with low variance during training. In contrast, generalist models such as OpenVLA and π_0 require significantly more training iterations to attain comparable accuracy and exhibit greater variance, which can be attributed to their large model capacity. Notably, although DP achieves lower loss by fitting directly to noise, it still demands more training steps to generate coherent actions, even after loss convergence.
- *Full Fine-tuning.* These fully fine-tuned VLA models are evaluated on three tasks: (1) out-of-distribution (OOD) object manipulation, (2) spatial relationship reasoning and (3) multi-object pick-and-place tasks. In task (1), both NORA and OpenVLA succeed, while SpatialVLA fails due to incorrect affordance point estimation. In task (2), NORA correctly follows instructions, while OpenVLA fails and SpatialVLA exhibits unstable performance. In task (3), NORA achieves successful execution while other models fail to complete the task reliably.

Sim2Real Adaption Performance. We compare model performance on simulation benchmarks and real robot deployments. A significant drop in performance is observed during transfer from simulation to the real world (Table 3; Appendix Table 4). The simulation benchmark includes 30 procedurally-generated, disentangled tasks requiring nuanced spatial reasoning (*LIBERO-Spatial*), object understanding (*LIBERO-Object*), and goal interpretation (*LIBERO-Goal*), as well as 10 long-horizon, entangled tasks (*LIBERO-Long*).

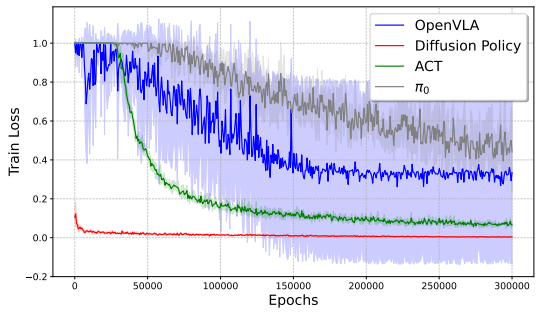

Figure 5: Results for partial fine-tuning of VLA models including OpenVLA and π_0, alongside results from training Diffusion Policy and ACT (Action Chunking Transformer) from scratch on our dataset. VLA models require more training epochs to converge and exhibit higher variance in performance.

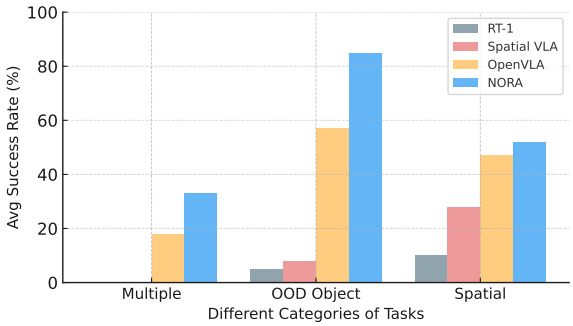

Figure 6: Success rates of fully fine-tuned VLA models on out-of-distribution object manipulation (OOD Object), spatial relationship reasoning (Spatial) and multi-object pick-and-place (Multiple) tasks. NORA achieves the highest performance.

Robustness to Perturbations. To evaluate robustness, we introduced distractor objects into the environment. As shown in Table 5, both OpenVLA and NORA exhibit substantial performance degradation in the presence of these perturbations, highlighting their sensitivity to novel conditions.

Key Takeaways. Current VLA models still face significant limitations in the following areas:

- *Adaptation and Generalization.* A generic robotic policy is expected to quickly adapt to datasets with distributional shifts. However, according to the partial fine-tuning results, due to the large model capacities and the limited size of task-specific datasets, these VLA models failed to achieve fast adaptation. While full fine-tuning offers improved performance, it requires extensive data and long training time, which are impractical for many real-world scenarios.
- *Robustness.* Robustness to distribution shifts (without finetuning) is a critical challenge. Results reveal substantial performance degradation both when encountering unseen objects and during sim-to-real transfer, highlighting the fragility of current VLA models in dynamic and unpredictable environments.

These findings suggest that while VLA models hold promise, they have limitations in data efficiency, adaptation speed, and robustness to make them reliable for real-world robotic applications.

5 Constraints and Future Directions

Despite the promise of foundation models for enabling embodied agents to perform daily tasks, the following critical constraints still hinder their widespread deployment:

Data Scarcity. In contrast to natural language datasets, which are readily sourced from internet, robotic datasets are significantly more expensive due to high hardware costs and intensive labor during data acquisition. A promising direction for future research is developing more data-efficient models. In addition, exploring high-fidelity simulation environments and developing robust sim-to-real transfer techniques could mitigate data scarcity.

Limited Generalization Capability. A key limitation of current VLA models is their limited ability to generalize to out-of-distribution concepts that were not well-represented during training, which is a consequence of the aforementioned data scarcity. Many models depend heavily on large-scale paired datasets, which often exhibit biases and limited diversity in aspects such as camera viewpoints, lighting conditions and specific robotic embodiments. This results in fragile performance when deployed in real-world or domain-specific scenarios. Furthermore, these models struggle with fine-grained spatial reasoning and temporal understanding, hindering them from accurately aligning language with complex visual scenes or dynamic events.

Efficient Inference. Deploying large-scale models on robotic platforms introduces significant computational challenges, primarily in inference speed and GPU RAMs. This underscores the importance of smaller models that can efficiently generate actions without significant performance degradation. This issue is particularly pronounced in autoregressive models, and diffusion models are less affected.

Limitations

When evaluating the generalization capabilities of Vision-Language Alignment (VLA) models, this paper primarily focuses on different tasks without extensively addressing their generalization across varying robot morphologies. Although relatively few works have specifically targeted this aspect, it remains a significant challenge in deploying VLA models in real-world applications. Robots with different morphologies, such as bimanual manipulators, humanoid robots, and autonomous vehicles, require distinct operational protocols and safety considerations. The absence of a generic robot policy that can adapt seamlessly across diverse morphologies limits the practical generalization potential of VLA models and hinders their deployment as universal robotic policies.

In addition, this paper does not explore the ability of VLA models to ground instructions involving open-ended or ambiguous commands. Current VLA models are largely trained on curated datasets, which allow them to learn mappings from instructions to specific actions. However, this reliance constrains their ability to truly understand instructions at the semantic level. As a result, when facing out-of-distribution or vague commands, these models often struggle to infer reasonable actions. Addressing this limitation will require integrating more advanced instruction-understanding modules into the VLA pipeline to improve their robustness in handling ambiguous or under-specified input.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-VL Technical Report. *Preprint*, arXiv:2502.13923.

Tirthankar Bandyopadhyay, Fletcher Talbot, and 1 others. 2024. Demonstrating Event-Triggered Investigation and Sample Collection for Human Scientists using Field Robots and Large Foundation Models. In *Robotics: Science and Systems*.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, and 1 others. 2025. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*.

Kevin Black, Noah Brown, and 1 others. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*.

Anthony Brohan, Noah Brown, and 1 others. 2023. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. 2025. Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2502.14420*.

Cheng Chi, Zhenjia Xu, and 1 others. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*.

Shichao Fan, Quantao Yang, Yajie Liu, Kun Wu, Zhengping Che, Qingjie Liu, and Min Wan. 2025. Diffusion trajectory-guided policy for long-horizon robot manipulation. *arXiv preprint arXiv:2502.10040*.

Jing Gao, Ning Cheng, and 1 others. 2024. Transformer in touch: A survey. *arXiv:2405.12779*.

Google DeepMind Gemini Robotics Team. 2025. Gemini Robotics: Bringing AI into the Physical World. *Preprint*, arXiv:2503.20020.

Marc Glocker, Peter Hönig, Matthias Hirschmanner, and Markus Vincze. 2025. LLM-Empowered Embodied Agent for Memory-Augmented Task Planning in Household Robotics. *Preprint*, arXiv:2504.21716.

Google DeepMind. 2024. Gemini: Our largest and most capable ai models yet. Accessed: 2025-05-17.

Aaron Grattafiori, Abhimanyu Dubey, and others Jauhri. 2024. The Llama 3 Herd of Models.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, and 1 others. 2025. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, and 1 others. 2025. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*.

Yunfan Jiang, Agrim Gupta, and 1 others. 2023. Vima: robot manipulation with multimodal prompts. In *ICML*.

Moo Jin Kim, Karl Pertsch, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv:2406.09246*.

Jinming Li, Yichen Zhu, Zhiyuan Xu, Jindong Gu, Minjie Zhu, Xin Liu, Ning Liu, Yaxin Peng, Feifei Feng, and Jian Tang. 2024a. MMRo: Are Multimodal LLMs Eligible as the Brain for In-Home Robotics?

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, and 1 others. 2024b. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*.

Xiaoqi Li, Mingxu Zhang, and 1 others. 2024c. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *CVPR*.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, and 1 others. 2025. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*.

Shilong Liu, Zhaoyang Zeng, and 1 others. 2024a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*.

Songming Liu, Lingxuan Wu, and 1 others. 2024b. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv:2410.07864*.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. DeepSeek-VL: Towards Real-World Vision-Language Understanding. *Preprint*, arXiv:2403.05525.

Octo Model Team, Dibya Ghosh, and 1 others. 2024. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*.

Abby O'Neill, Abdul Rehman, and 1 others. 2024. Open x-embodiment: Robotic learning datasets and rt-x models. In *ICRA*.

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

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and 1 others. 2025. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*.

Nikhila Ravi, Valentin Gabeur, and 1 others. 2025. SAM 2: Segment anything in images and videos. In *ICLR*.

Tianhe Ren, Shilong Liu, and 1 others. 2024. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv:2401.14159*.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. 2024. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *Robotics: Science and Systems*.

Qi Sun, Pengfei Hong, and 1 others. 2024. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv:2412.11974*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Junjie Wen, Minjie Zhu, and 1 others. 2024. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv:2412.03293*.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. 2025. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. 2024. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In *Robotics: Science and Systems XX*.

Fengyu Yang, Chao Feng, and 1 others. 2024. Binding touch to everything: Learning unified multimodal tactile representations. In *CVPR*, pages 26340–26353.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. 2025. Up-vla: A unified understanding and prediction model for embodied agent. *arXiv preprint arXiv:2501.18867*.

Jialiang Zhao, Yuxiang Ma, and 1 others. 2024. Transferable tactile transformers for representation learning across diverse sensors and tasks. In *CoRL*.

Ruijie Zheng, Yongyuan Liang, and 1 others. 2025. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *ICLR*.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and 1 others. 2025. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*.

Brianna Zitkovich, Tianhe Yu, and 1 others. 2023. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*.

A Details for case studies on robotic manipulation

A.1 VLA Evaluation Dataset Construction

To analyze the convergence behavior of various VLA models under distribution shift, we constructed a custom cluttered tabletop environment using a UR5 robotic arm equipped with a wrist-mounted RealSense RGB-D camera, which is distinct from any existing settings in the Open-X-embodiment dataset. Demonstration data for a screwdriver-picking task among distractor objects was collected using a SpaceMouse for teleoperation[1].

We collected 163 demonstration episodes, each beginning with a randomized initial robot pose followed by an attempt to grasp the screwdriver. OpenVLA and π_0 were partially fine-tuned on this dataset, while Diffusion Policy (DP) and Action Chunking Transformer (ACT) were trained from scratch.

Due to the limited size of our custom dataset, full fine-tuning of RT-1, OpenVLA, SpatialVLA, and NORA was performed using the Open-X-embodiment and LIBERO datasets. We evaluated model performance on both a real-world WidowX robotic platform and the LIBERO simulation benchmark.

A.2 Fine-tuning details for VLA

Partial fine-tuning was conducted on a single NVIDIA A6000 GPU (48 GB VRAM) over a period of three days. To ensure a fair comparison, a batch size of 1 was used across all models. The results are presented in Fig. 5.

Full fine-tuning of RT-1, OpenVLA, SpatialVLA, and NORA was conducted on a compute node equipped with 8×H100 GPUs. The fine-tuned models were evaluated on 9 diverse real-world manipulation tasks, as shown in Fig. 7. Success rates are summarized in Table 4, demonstrating NORA's superior policy generation capabilities across three task categories: out-of-distribution object grasping, spatial reasoning, and multi-object manipulation.

[1]The dataset will be released upon acceptance.

A.3 Impact of Action Chunking

A.3.1 Action Chunking Performs Poorly on WidowX.

To investigate the effectiveness of action chunking, we selected NORA-LONG and SpatialVLA for evaluation. Tasks were chosen from three categories: (1) "`put the carrot in the pot`," (2) "`put the red bottle and hamburger in the pot`," and (3) "`put the pink toy at the right corner`." In initial experiments, all predicted actions (5 actions for NORA-LONG, 4 actions for SpatialVLA) were executed sequentially without replanning. This frequently caused the WidowX robot to crash into the environment due to the accumulation of overly large movements.

Subsequently, we modified the execution policy to only perform the first action in each predicted chunk. This adjustment resolved the collision issue and NORA -LONG achieved an 80% success rate on the "`put the carrot in the pot`" task. However, on multi-object pick-and-place tasks, NORA-LONG consistently stopped after placing the first object, resulting in a 0% final success rate. For the spatial reasoning task, NORA-LONG achieved a 70% success rate on "`put the pink toy at the right corner`."

A.3.2 Action chunking improves performance in simulation.

We hypothesize that action chunking is more effective at higher control frequencies. For example, Diffusion Policy generates commands at 10 Hz, which are then interpolated to 125 Hz for execution. Similarly, OpenVLA-OFT+ employs action chunking and shows improved performance in real-world ALOHA tasks, which run at 25 Hz.

Since our real robotic platforms do not support high-frequency control, we tested this hypothesis in the LIBERO simulation environment (20 Hz). We fine-tuned both NORA and NORA-LONG on this benchmark with an action chunk size of 5, producing two variants: NORA-finetuned-AC and NORA-Long-finetuned.

Results show that NORA-finetuned-AC significantly outperforms NORA-finetuned across all LIBERO benchmarks, with a higher average success rate. Notably, NORA-Long-finetuned outperforms all baseline models (see Table 3), highlighting the benefits of pretraining with action chunking and its transferability to long-horizon tasks. However, it is important to note that LIBERO is a simu-

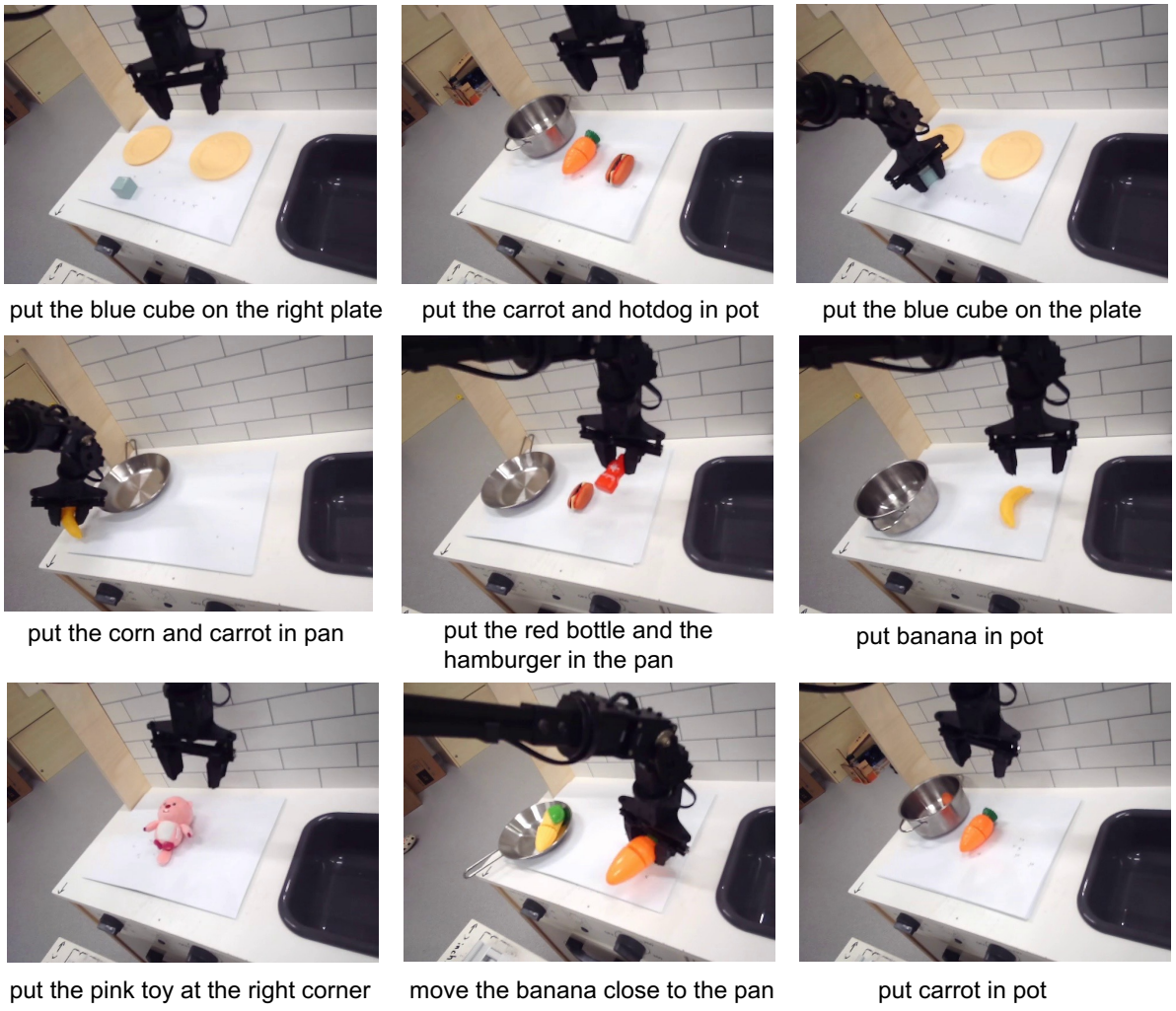

Figure 7: Real-world robot environments and task setups. We evaluate these models across 9 diverse tasks to assess its instruction understanding, spatial reasoning, and multi-task motion planning capabilities.

Category	Task	RT-1	OpenVLA	SpatialVLA	NORA
Multiple objects	Put the red bottle and the hamburger in the pan	0	20	0	**40**
	Put the carrot and hotdog in pot	0	0	0	**30**
	Put the corn and carrot in the pan	0	**30**	0	**30**
OOD object	put carrot in pot	0	80	20	**90**
	Put banana in pot	1	40	0	**90**
	Put the blue cube on the plate	0	50	0	**70**
Spatial	Put the pink toy at the right corner	0	**60**	30	**60**
	Put the blue cube on the right plate	0	**30**	0	20
	Move the banana close to the pan	30	50	50	**80**
Average		4.4	40	11.1	**56.7**

Table 4: Task performance comparison across different categories and models.

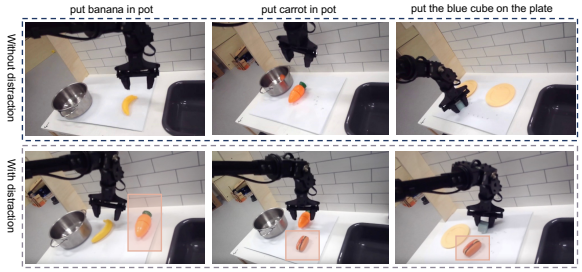

Figure 8: Comparison of tasks with and without distraction.

Table 5: Average Success Rate (%) without (w/o) and with (w/) Distractors

Model	w/o Distractors	w/ Distractors
OpenVLA	56.7	50
NORA	83.3	56.7

lation environment and may not reflect real-world performance at high control frequencies.

A.4 Robustness to Disturbance

To evaluate robustness, we selected three straightforward tasks (shown in Fig. 8) and introduced distractor objects into the environment. Initially, both OpenVLA and NORA performed well. However, their success rates declined significantly with the introduction of distractions. This highlights the sensitivity of current VLA models to out-of-distribution disturbances. The average success rates across the three tasks are presented in Table 5, while the detailed number of successful executions out of 10 trials is summarized in Table 6.

TASK	OpenVLA	NORA
without Distraction		
put carrot in pot	8	9
put banana in pot	4	9
put the blue cube on the plate	5	7
with Distraction		
put carrot in pot	6	8
put banana in pot	6	4
put the blue cube on the plate	3	5

Table 6: Comparison of task performance between OpenVLA and NORA under conditions with and without distraction. Each value denotes the number of successful executions out of 10 trials.

A.5 Modularized Testbed for Evaluating VLMs

To facilitate the evaluation of different VLMs in robotic manipulation, we developed a voice-controlled testbed using a UR5 robotic arm[2]. The system architecture, shown in Fig. 9, comprises the following five modules:

- **Speech Transcription:** Powered by Microsoft Azure's speech recognition service.
- **Task Decomposition:** Based on GPT-3.5 and GPT-4 using prompting paradigms adapted from ChatGPT for Robotics.
- **Object Detection:** Utilizes GroundingDINO and OWL-ViT for object detection.
- **Object Segmentation:** Employs Segment Anything Model (SAM) and FastSAM for segmenting detected objects.

[2]The source code will be released upon acceptance.

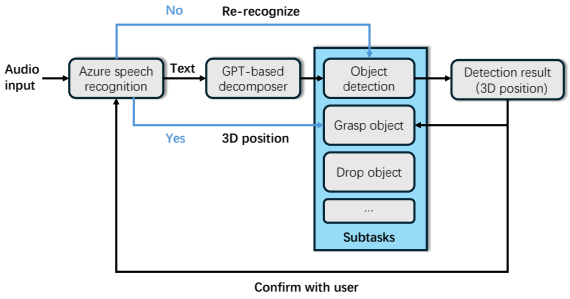

Figure 9: The system architecture of the testbed for VLMs.

Words	*Positional*: left, right, between, beside, near, far, front, behind	*Directional*: aligned with, perpendicular to
Instruction	hand over the screwdriver [on the left of] the red ball.	pass me the screwdriver [aligned with] the marker.

Table 7: Template words and corresponding examples of generated relation-based instructions for case studies.

- **Manipulation:** Low-level actions are generated by GraspAnything or GraspNet.

This modular testbed enables rapid integration and benchmarking of different models within a real robotic system.

B Details for Case Studies on Instruction Grounding

B.1 VLM Evaluation Dataset Construction

To evaluate the capabilities of VLMs, we developed a dataset specifically designed to test their ability to identify objects based on explicit attributes, explicit location relations, and functions. Additionally, the dataset includes multi-turn questions that refer to more than one object, requiring VLMs to ask clarifying questions to identify the correct object.

- **Explicit Attributes.** In this category, instructions prompt VLMs to identify objects belonging to a category with multiple instances, where each instance can be uniquely identified by explicitly mentioned attributes. For example, in Figure 3, the beige mug and the gray mug are included because they are unique when described with attributes. However, objects like the black mug or scissors are excluded. This is because there are two identical black mugs, making them non-unique, and there is only one pair of scissors, which does not require attributes for identification.

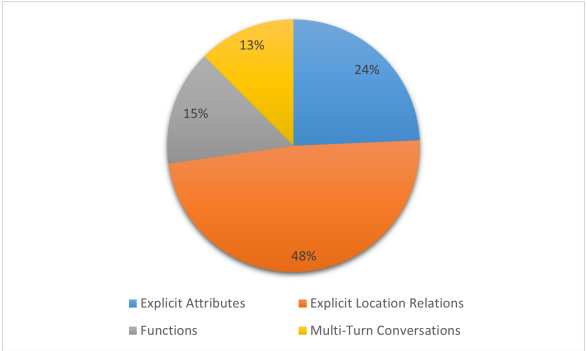

Figure 10: Distribution of Instruction Types

- **Explicit Spatial Relationships.** In this category, instructions describe objects by their spatial relationships to other objects in the image. We ensure that each referenced object is unique within the image. For example, the measuring cup to the right of the screwdriver uniquely identifies the object. These instructions are designed to test the VLMs' ability to comprehend and resolve location-based relationships.
- **Functions.** Here, objects are not explicitly mentioned by name or attributes but are instead described by their functions. This category evaluates the VLMs' ability to infer the correct object based on its use. For example, the dataset includes instructions referring to objects like scissors, screwdrivers, and rulers based on their respective functions.
- **Multi-Turn Conversations.** This category involves instructions referencing multiple objects in the same image. For example, Figure 3 shows two black mugs. In such cases, VLMs are expected to ask clarifying questions to gather more specific information to identify the intended object.

To ensure high-quality data, we employed a human-in-the-loop process to verify the outputs of VLMs and LLMs:

- **Initial Object Identification**: We used GPT-4o to identify objects in an image and referring them by type, explicit attributes, and detailed location relations.
- **Human Verification.** The authors of this paper reviewed and modified the outputs to ensure their correctness.
- **Instruction Generation.** After verification, GPT-4 was tasked with generating simple, clear instructions for different objects.

	Easy			Medium			Hard		
	im	attr	rel	im	attr	rel	im	attr	rel
VLM+GPT-4	0.05	0.516	0.131	0.01	0.336	0.186	0	0.318	0.174
GPT-4o-0513	0.850	1.000	0.778	0.819	0.948	0.680	0.901	0.697	0.469
GPT-4o-mini	0.750	0.717	0.550	0.764	0.771	0.596	0.750	0.382	0.248
GPT-4	0.650	0.750	0.598	0.750	0.737	0.662	0.625	0.417	0.455
Qwen2-VL-72B	0.800	0.917	0.830	0.792	0.756	0.738	0.875	0.700	0.529
Llama-3.2V-90B	0.750	0.850	0.704	0.708	0.853	0.711	0.875	0.491	0.521
Llama-3.2V-90B-Q4	0.800	0.667	0.598	0.625	0.719	0.554	0.542	0.464	0.300
Llama-3.2V-11B	0.650	0.667	0.631	0.764	0.710	0.556	0.833	0.536	0.342
Llama-3.2V-11B-Q4	0.650	0.567	0.502	0.694	0.757	0.555	0.542	0.498	0.450

Table 8: Performance Metrics Across Easy, Medium, and Hard Tasks. im: implicit instructions. attr: explicit attributes. rel: relative relations.

- **Final Review.** These instructions underwent another round of verification to ensure clarity and accuracy.

As a result, we have created a high-quality dataset consisting of 30 images and 473 instructions, with a detailed breakdown of each instruction type presented in Fig. 10.

B.2 Failure Cases of Specialist VLM Pipelines

Grounding DINO, despite popular for zero-shot detection, is not robust in open scenes. It successfully detected "blue ball" while failed to detect "ball", indicating its reliance on visual features. Similarly, featureless metal cans pose a great challenge for Grounding DINO, which were almost omitted in the detection results. For complex instruction grounding, Grounding DINO and GPT-4 were chained together to "guess" the target by the LLM based on the candidate bounding boxes. The failure cases were illustrated in the Fig. 11 and Fig. 12.

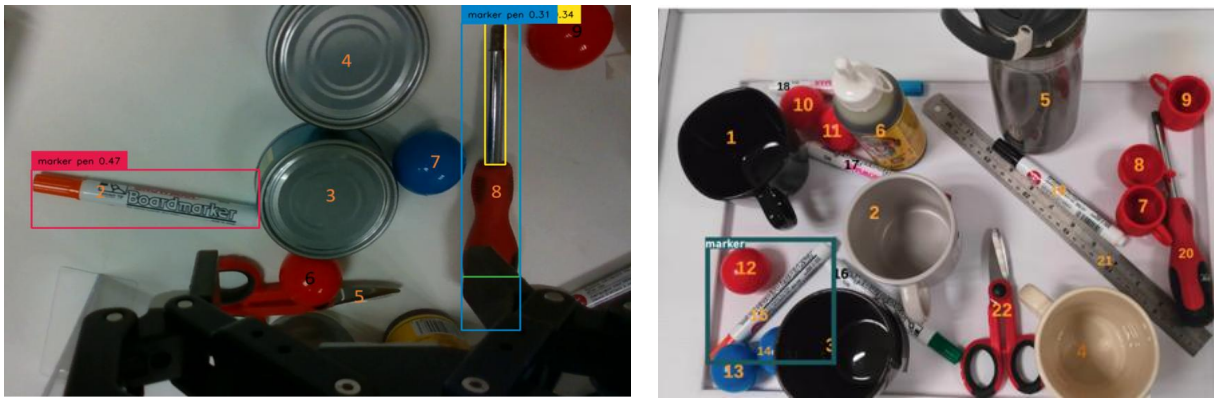

Figure 11: Examples of Instruction Grounding. (a) “the marker on the left”, (b) “the marker aligned with the ruler”.

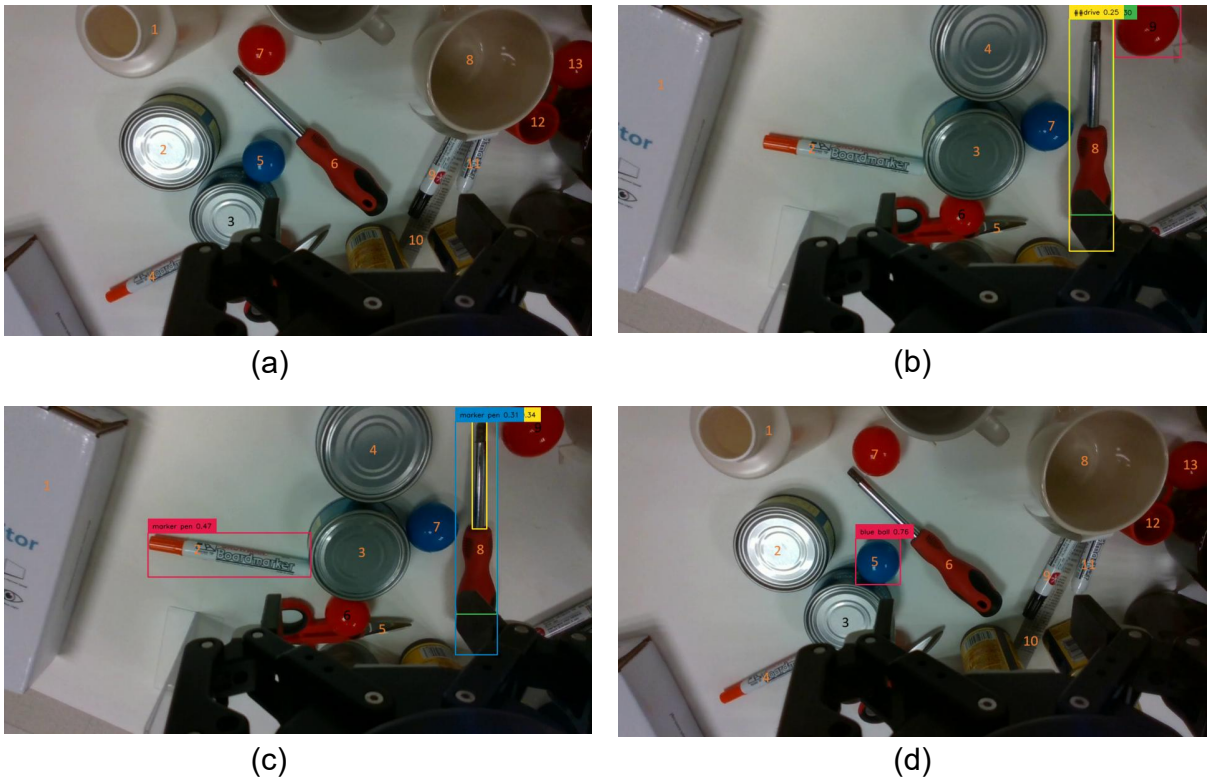

Figure 12: Examples of Object Grounding. (a) “ball”, (b) “screwdriver”, (c) “marker pens”, (d) “blue ball”.