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

Xiuchao Sui[1*] **and Daiying Tian**[1*] **and Qi Sun** [2] **and Ruirui Chen**[1] **and Dongkyu Choi** [1] **and Kenneth Kwok** [1] **and Soujanya Poria** [2]
[1]IHPC, Agency for Science, Technology and Research, Singapore
[2]Nanyang Technological University, Singapore

Abstract

Foundation models (FMs) are increasingly applied to bridge language and action in embodied agents, yet the operational characteristics of different integration strategies remain under-explored—especially for complex instruction following and versatile action generation in changing environments. We investigate three paradigms for robotic systems: end-to-end vision-language-action models (VLAs) that implicitly unify perception and planning, and modular pipelines using either vision-language models (VLMs) or multimodal large language models (MLLMs). Two case studies frame the comparison: instruction grounding, which probs fine-grained language understanding and cross-modal disambiguation; and object manipulation, which targets skill transfer via VLA finetuning. Our experiments reveal trade-offs in system scale, generalization and data efficiency. These findings indicate design lessons for language-driven physical agents and point to challenges and opportunities for FM-powered robotics in real-world conditions.

1 Introduction

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

However, realizing the promise of *language-grounded autonomy* in deployable systems remains highly challenging. Robots must (i) map ambiguous instructions to the physical world (*instruction grounding*), (ii) execute reliably across novel objects, scenes, and morphologies (*generalizable execution*), and (iii) achieve these goals with limited data (*efficient adaptation*). How effectively different FM integration strategies address these competing requirements remains under-explored (Fig. 1).

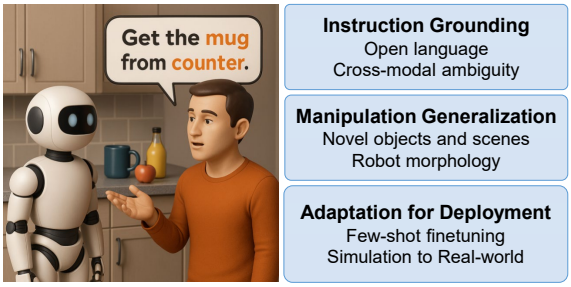


Figure 1: Key challenges of FMs in embodied robotics.

This work offers an empirical study of three integration paradigms: end-to-end VLAs mapping language and vision to actions, MLLM agents orchestrating perception and control, and modular VLM pipelines pairing perception-specialist FMs with task-specific planners (Fig.2; Table1).

We evaluate these paradigms through tabletop case studies that expose their complementary strengths and limitations. Two task categories are considered: complex instruction grounding, probing fine-grained understanding and cross-modal disambiguation (Sec.3); and object manipulation, assessing skill transfer after VLA fine-tuning under distribution shifts, complemented by comparative and ablation analyses (Sec.4).

Our instruction grounding experiments reveal distinct trade-offs across integration strategies. VLM pipelines emphasize interpretability and efficiency, but fall short of peak performance. They struggle with complex instruction grounding yet achieve moderate object grounding—using less than 1% of the parameters required by MLLMs.

*Equal contribution. Corresponding authors. {Sui_Xiuchao, Tian_Daiying}@a-star.edu.sg

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1324–1340
November 4-9, 2025

In contrast, MLLMs generalize better on complex instructions but incur substantially higher inference costs, with smaller reasoning-focused models at times outperforming larger ones. VLAs, with tightly coupled perception-to-action pathways, streamline action generation but make it difficult to isolate and assess grounding performance. To probe this gap, we evaluate their perception heads as reference points, which also surfaces open questions for VLA architecture design.

While MLLMs perform well on grounding tasks, their scale limits practical deployment on robotic platforms. Quantization is often adopted as a simple remedy, yet instruction-dependent behaviors emerge, with certain reasoning capabilities degrading disproportionately under naïve compression. This underscores the need for fine-grained quantization strategies. Taken together, our findings clarify the trade-offs between model scale and performance and provide practical guidance for building robotic systems under real-world constraints.

Our object manipulation experiments investigate the skill-adaptation capabilities of VLAs in both real-world and simulated settings. We examine how these models transfer manipulation skills under distribution shifts, evaluating their training dynamics, generalization, and robustness. Results reveal fragile training and slow adaptation in generalist policies, such as OpenVLA (Kim et al., 2024), compared to task-specific counterparts, underscoring the challenges of real-world deployment."

In summary, our main contributions are:

- We analyse three FM integration paradigms on shared embodied tasks designed to probe the capabilities and trade-offs of FMs.
- We release a dataset and code[1] for evaluating instruction grounding and object manipulation, covering cross-modal reasoning and skill adaptation under varied layouts.
- We providing timely insights into state-of-the-art VLAs and MLLMs, investigating their capabilities and failure modes and distilling practical trade-offs to guide practitioners in selecting FM stacks for language-driven embodied agents.
- We also release a complete end-to-end claw-machine robot system as a real-world FM integration demo[2].

[1] https://github.com/xiuchao/InstructionGrounding
[2] https://github.com/HRItdy/claw_machine

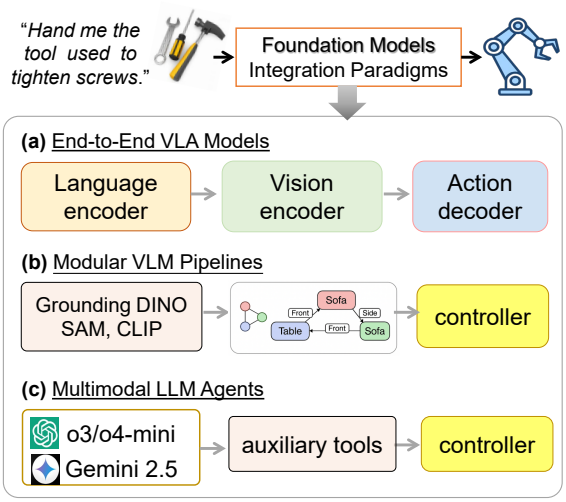


Figure 2: Three FM integration strategies for embodied robotics, highlighting distinct interfaces between language, perception, and control.

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 VLA Models

Definition. VLAs 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.

Autoregressive Models. These models generate actions step by step, with each action conditioned on the current perceptual input and previously generated outputs. These models typically encode visual observations and language into a shared latent space, then employ a transformer-based decoder to autoregressively predict low-level control tokens, such as joint angles, end-effector displacements and gripper states, which are typically obtained by discretizing continuous motor signals into token sequences.

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.

RT-1 (Brohan et al., 2023) and RT-2 (Zitkovich et al., 2023) demonstrated the effectiveness of large-scale pretraining for task generalization. Building on these scaling successes, OpenVLA (Kim et al., 2024) became a landmark open-source effort, combining Llama-2 with DinoV2 and SigLIP, trained on the large-scale Open-X-Embodiment dataset (Collaboration et al., 2024). Subsequent works such as Emma-X (Sun et al., 2024), NORA (Hung et al., 2025) and TraceVLA (Zheng et al., 2025), further advanced this line through dataset quality, stronger backbones and improved spatiotemporal prompting.

Subsequent models further incorporate *richer modalities* such as tactile sensing (Yang et al., 2024; Zhao et al., 2024), *structural priors* including spatial representations distilled from VLMs (Gemini Robotics Team, 2025) and 3D spatial relationships in SpatialVLA (Qu et al., 2025), and refinements to the *embodied latent space*, as in UniVLA (Bu et al., 2025), which learns task-specific latent representations.

Diffusion-based Models. They generate actions by progressively denoising trajectories from noise, modeling the distribution of future actions as a whole rather than step by step. Like autoregressive VLAs, these models encode visual observations and language into a latent representation, but instead apply a diffusion process that iteratively refines action sequences into feasible trajectories. Diffusion-based models trade inference speed for greater trajectory coherence.

Diffusion Policy (Chi et al., 2023) pioneered the use of diffusion models for visuomotor policy learning, showing improved global consistency and robustness compared to autoregressive baselines. Octo (Team et al., 2024) applies conditional diffusion decoding for action sequence prediction. Subsequent models further scales diffusion heads into larger, dedicated policy modules (Reuss et al., 2024; Wen et al., 2024; Li et al., 2024b), and integrate trajectory-level guidance to improve long-horizon planning (Fan et al., 2025). Notable models like π_0 (Black et al., 2024) combine pretrained MLLM (PaliGemma-3B) with flow-matching-based action experts to produce continuous, high-frequency control across robot embodiments. $\pi_{0.5}$ (Intelligence et al., 2025) builds on this by co-training across diverse environments and data modalities to improve generalization to new settings.

Beyond architectural advances, diffusion-based models increasingly targets diverse embodiments, extending from single-arm manipulation to bimanual systems and humanoid robots. Representative examples include RDT-1B (Liu et al., 2024b), DexVLA (Wen et al., 2025), and GR00T N1 (Bjorck et al., 2025), which demonstrate the scalability of FM-driven action generation to more complex morphologies.

Strengths and Limitations. Leveraging large-scale pretraining, VLAs hold the potential to generalize across diverse manipulation tasks and robot morphologies. Yet progress is constrained by the scarcity of high-quality, diverse robotic datasets, which limits both scale and coverage. Pretraining can also propagate biases from training distributions, causing degraded performance on novel tasks, in unseen environments, or with unfamiliar embodiments. Despite their promise, VLAs remain brittle in real-world deployment, as highlighted in our *Object Manipulation* case study (Section 4), underscoring the need for better data curation, bias mitigation, and efficient adaptation strategies such as few-shot and continual learning.

2.2 Modular VLM Pipelines

Definition. In this paradigm (Fig. 2b), perception is handled by a *specialist* VLM that outputs symbolic scene information, typically grounded 2D/3D bounding boxes, segmentation masks, or referring expression pointers. A downstream plan-

ner 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 directly inspected; (ii) *efficiency*, models typically contain 100M$\sim$600M parameters, only about 1% $\sim$ 6% the size of representative 10B-scale MLLMs (Grattafiori et al., 2024). However, they are limited by (i) *interaction rigidness* compared to more flexible MLLMs, and (ii) *pipeline brittleness* where perception errors propagate without mitigation (Fig. 2b; Table 1). Their effectiveness depends on robust open-vocabulary grounding—precisely the capability highlighted in our *Instruction Grounding* case study (Section 3).

2.3 Multimodal LLM Agents as Orchestrators

Definition. In this paradigm (Fig. 2c), MLLMs take raw user utterances, selectively invoke vision tools (*e.g.*, a detector or depth estimator) via function calls, reason over their outputs in context, and issue high-level action primitives to a low-level controller. An MLLM agent thus places a large tool-calling language model at the center of the control loop, acting as a *cognitive hub* that binds perception and control through natural language.

Representative Systems. MLLMs are playing increasingly important roles in robotics. Gemini Robotics (Gemini Robotics Team, 2025) integrates perception, spatial reasoning, and trajectory synthesis into a single Gemini-2.0 backbone (Google DeepMind, 2024), serving as an embodied brain. ManiLLM (Li et al., 2024c), in a similar spirit, leverages the common-sense and reasoning capabilities of MLLMs by fine-tuning adapter modules with a chain-of-thought training paradigm, enabling accurate pose prediction and precise manipulation. These works illustrate the emerging trend of MLLMs shifting toward the role of a *cognitive hub* in robot systems. Hub-LLaMA (Glocker et al., 2025) further builds a modular agent-orchestration system for household object management, using LLaMA 3.2 Vision (Grattafiori et al., 2024) for open-vocabulary perception to ground task plans, though its limitations are not discussed.

Somewhat related to our work, (Li et al., 2024a) evaluates the suitability of MLLMs as a "brain" for in-home robotics, providing a benchmark that compares models across perception, visual reasoning, and task planning. The benchmark included a few MLLMs available at the time, while newer releases were not covered—illustrating the rapid pace of progress in this area.

Strengths and Limitations. MLLMs 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*, posing challenges for deployment—particularly on mobile robotic platforms. We further examine the capability limits and trade-offs of MLLMs (Section 3).

3 Case Studies on Instruction Grounding

Natural language *instruction grounding* translates user intent into actionable goals within a visual scene. Our case study evaluates grounding performance under challenging cross-modal disambiguation, highlighting trade-offs and offering guidance for efficient deployment.

Benchmark Dataset. To isolate grounding ability from general vision priors, we design carefully controlled scenarios using household objects placed on a tabletop. These objects are widely represented in FM training corpora, while the tabletop setup minimizes variation in lighting and camera angles—ensuring the evaluation primarily reflects grounding performance.

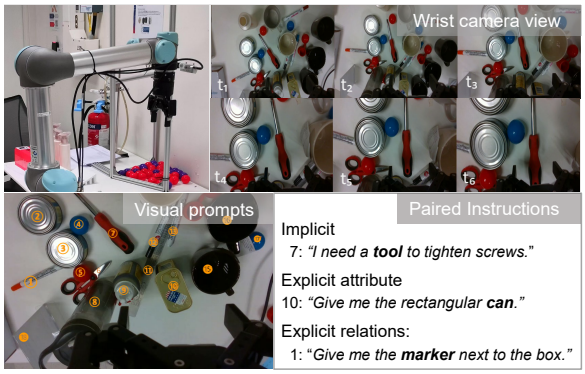


Figure 3: Experimental setup for two case studies in a cluttered tabletop environment. The top row shows egocentric video data collected for the manipulation case study. The bottom row is an example setup for the instruction grounding task, including an annotated visual prompt paired with complex instructions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

We curated a new Instruction Grounding benchmark (Fig. 3) consisting of images with multiple household objects, each tagged with numbers as visual prompts and paired with instructions targeting visual commonsense and cross-modal disambiguation. For example, "*pick up the red-capped marker*" requires attribute reasoning to choose among markers, while "*grasp the cup in front of the screwdriver*" tests spatial reasoning (Appendix Table 4). These tasks highlight how grounding errors can lead to execution failures in embodied systems.

Zero-Shot Object Grounding. We start with a necessary first question for instruction grounding: *Can FMs reliably recognize objects in cluttered, open-world scenes?* Table 2 highligh key observations and point to opportunities for VLA design.

- Despite their popularity in modular pipelines, GroundingDINO achieves around 0.3-0.4 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 MLLMs which embodied large volume of visual commonsense (Appendix Fig. 10).
- Gemini 2.5-Pro and GPT-5 achieve top performance with an average score of 0.82, followed by Gemini 2.0-Flash and GPT-4.5. Open-source models continue to lag behind, with LLaMA 3.2-Vision 90B reaching 84% of the performance of top proprietay models.
- Small- and medium-scale models (e.g., Gemma-27B, Phi-Vision) generally fall below the specialist threshold, underscoring their limitations for fine-grained object grounding in open scenes. Notable exceptions include GPT5-mini (0.754) and QwenVL-3B (0.543), which offer favorable speed–accuracy trade-offs, achieving reasonable performance without high API costs or heavy memory footprint of larger models.
- PaliGemma-3B in π_0 struggles to follow structured prompts and can only follow simpler instructions. Its grounding capability is highly limited, often detecting just a single object per scene. Reported accuracy is therefore approximate, based on a looser evaluation criterion. While QwenVL-3B adopted in NORA, reaches higher grounding accuracy, which may partially explain its stronger performance on out-of-distribution manipulation tasks (Appendix Table 6).
- Two *design questions* emerges: (i) how should perception modules be selected or adapted to improve instruction-following in embodied systems? and (ii) can compact, grounding-capable perception modules be distilled from large-scale models to support efficient deployment?

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
Proprietary MLLMs				
Gemini2.5-Pro-Exp	**0.904**	0.765	**0.793**	0.821
Gemini2.0-Flash	0.884	0.738	0.678	0.767
GPT-5-auto	0.881	**0.829**	0.760	**0.823**
GPT-5-mini	0.749	0.737	0.776	0.754
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 MLLMs				
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
Base MLLMs in VLAs				
PaliGemma-3B (π_0)	0.118	0.07	0.05	0.079
QwenVL-3B (NORA)	**0.519**	**0.535**	**0.577**	**0.543**

Table 2: Object grounding performance of specialist VLMs and MLLMs across cluttered scenes of varying complexities, with macro accuracy reported.

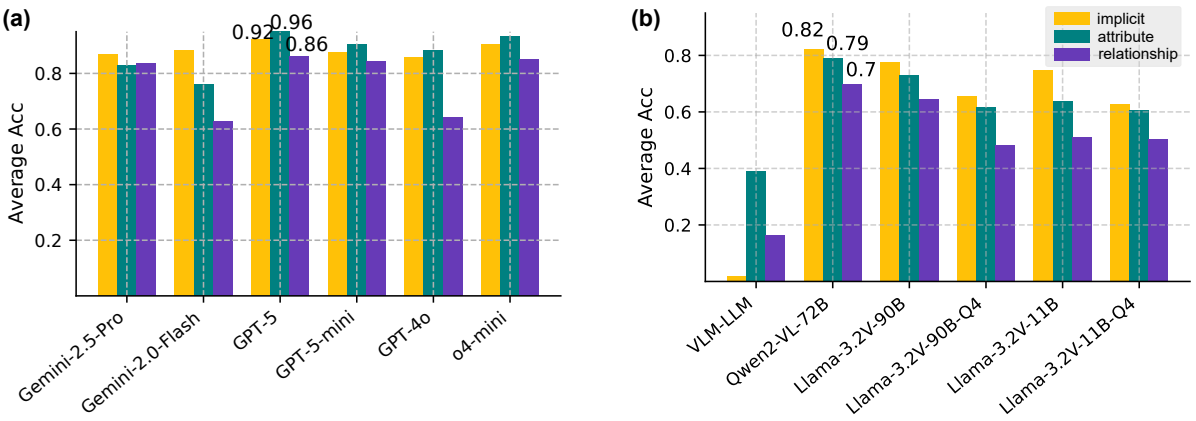


Figure 4: Performance of complex instruction grounding across modular VLM pipeline and MLLMs. Macro accuracy is reported across instruction types—implicit, attribute-based, and relationship-based. Subfigures show (a) proprietary models and (b) open-source models along with their Int4-quantized variants.

Zero-Shot Complex Instruction Grounding. This task is framed as a multiple-choice problem, where the model selects the correct object index in a cluttered scene given three types of natural language instructions: implicit, attribute-based, and relationship-based. Each type probes a distinct grounding challenge (Fig. 4). As a baseline, we evaluate a modular VLM pipeline in which GPT-4 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 pipeline struggles to select a screwdriver, lacking embedded affordance reasoning. In contrast, MLLMs perform well, reflecting strong visual commonsense. GPT-5 achieves 0.92 accuracy, while 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 (Appendix Table 5).

- *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. GPT-5, Gemini 2.5-Pro and o4-mini achieve accuracy above 0.80—demonstrating the benefits from embodied training data and strong reasoning capabilities. Notably, o4-mini is a medium-sized model, yet it outperforms larger models like GPT-4o 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.

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**

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.

fine-tuning, which leverages our curated probing dataset (Fig. 3) 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.

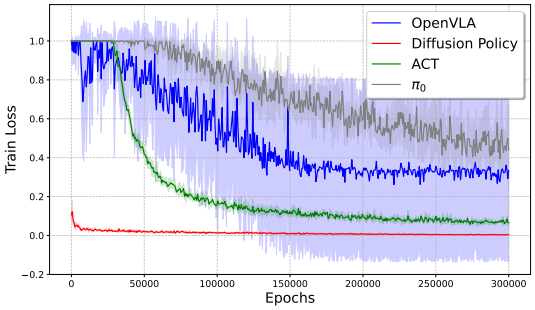


Figure 5: Partial fine-tuning results for VLAs (OpenVLA and π_0) compared with training Diffusion Policy (DP) and ACT from scratch on our dataset. VLAs require more epochs to converge and show higher performance variance.

Real-World Skill Adaptation. Our fine-tuning process consists of two stages: (1) To assess convergence on tasks beyond the coverage of common pretraining datasets, we collected a custom dataset (Appendix A.1) focused on screwdriver-picking in cluttered tabletop scenes. This task introduces both object- and scene-level distribution gaps, as screwdrivers and dense clutter are largely absent from Open-X-Embodiment (Collaboration et al., 2024). We used this dataset to partially fine-tune generalist VLA models, and to train compact task-specific models such as Diffusion Policy (DP) and Action Chunking Transformer (ACT) from scratch. (2) For full fine-tuning, we leveraged larger benchmarks—Open-X-Embodiment and the simulated LIBERO dataset (Liu et al., 2023)—to fully fine-tune RT-1, OpenVLA, SpatialVLA, and NORA, and compared their performance.

- *Partial Fine-tuning.* We observe that DP and ACT converge stably with low training variance (Fig. 5). In contrast, generalist models such as OpenVLA and π_0 require far more iterations to reach comparable accuracy and exhibit greater variance. This difference reflects model scale: compact task-specific policies like DP and ACT contain only tens of millions of parameters, whereas generalist policies such as OpenVLA (7B) and π_0 (3B) rely on billion-scale backones, making optimization slower and less stable. Notably, while DP attains lower loss by fitting directly to noise, it still requires additional training to produce coherent action sequences even after loss convergence.

- *Full Fine-tuning.* The fine-tuned VLAs are evaluated on three tasks: (1) out-of-distribution object manipulation, (2) spatial relationship reasoning, and (3) multi-object pick-and-place. In task (1), both NORA and OpenVLA succeed, whereas SpatialVLA fails due to incorrect affordance-point estimation. In task (2), NORA follows instructions correctly, while OpenVLA fails and SpatialVLA shows unstable performance. In task (3), only NORA executes the task successfully, while the other models fail to complete it reliably (Fig. 6).

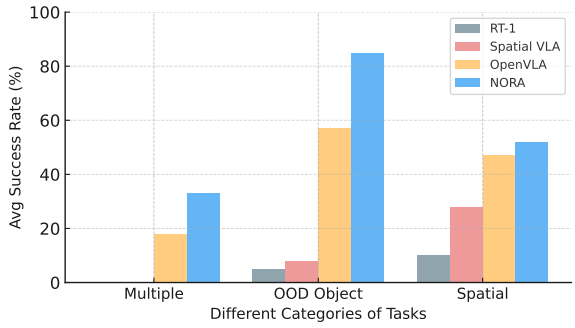


Figure 6: Success rates of fully fine-tuned VLAs on multi-object pick-and-place, out-of-distribution object manipulation, and spatial relationship reasoning tasks. NORA achieves the highest performance.

Simulated Skill Adaptation. We compare model performance on simulation benchmarks and real robot deployments (Table 3). The results show that the finetuned π_0 model outperforms all baselines across tasks, achieving the highest success rates on Spatial and Object tasks while maintaining strong performance on Goal and Long-horizon tasks. Furthermore, the ablation study of action chunking (AC) on NORA demonstrates that AC consistently enhances performance across most simulation tasks. Notably, a significant drop in performance is observed when transferring from simulation to the real world (Appendix Table 6).

Robustness to Perturbations. We assess robustness by introducing distractor objects into the environment. As shown in Appendix Table 8, both OpenVLA and NORA degrade substantially under these perturbations, underscoring their sensitivity to novel conditions.

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

- *Adaptation and Generalization.* A generic robotic policy is expected to adapt rapidly to datasets with distributional shifts. However, our partial fine-tuning results show that, given their large model capacities and the limited size of task-specific datasets, current VLAs fail to achieve efficient adaptation. While fine-tuning improves performance, it demands extensive data and prolonged training, which is usually impractical for many real-world scenarios.
- *Robustness.* Robustness to distribution shifts without finetuning remains a critical challenge. Results reveal substantial degradation when encountering unseen objects and during sim-to-real transfer, underscoring the fragility of current VLA models in dynamic and unpredictable environments.

These findings indicate that although VLAs hold significant promise, they remain constrained by poor data efficiency, slow adaptation, and brittle robustness. Bridging these gaps will require both algorithmic advances—such as more parameter-efficient adaptation, bias-resistant pretraining, and stronger perception backbones—and system-level improvements in data collection and augmentation. Addressing these limitations is crucial if VLAs are to evolve from research prototypes into reliable, deployable policies for real-world robots operating under uncertainty.

5 Constraints and Future Directions

Despite the promise of foundation models (FMs) for enabling embodied agents to perform daily tasks, several critical constraints still hinder their reliable deployment:

Data Scarcity. Unlike natural language data that abundantly available on the internet, robotic datasets are costly to collect due to hardware wear, safety risks, and labor-intensive demonstrations. A key direction is improving data efficiency through parameter-efficient adaptation, imitation from unlabeled interaction, and self-supervised pretraining. Complementary approaches include leveraging high-fidelity simulators and developing robust sim-to-real transfer pipelines to reduce reliance on large real-world collections.

Efficient Inference. VLAs place heavy computational burdens on robotic platforms, leading to bottlenecks in inference speed. The constraints motivate research on lightweight architectures and efficient decoding strategies that can sustain performance while satisfying the real-time requirements of embodied control.

Explainability and Safety. Most FMs lack explicit mechanisms for interpretability or safety guarantees—factors that are crucial for deployment in high-stakes or unstructured environments. These models may output confident but incorrect actions when faced with out-of-distribution inputs or adversarial perturbations. Moreover, without built-in constraints to enforce ethical and operational boundaries, VLAs risk misinterpreting ambiguous instructions in ways that compromise human intent or physical safety.

Limitations

For VLA generalization study, this work focuses primarily on task-level performance and does not extensively examine generalization across diverse robot morphologies. Yet this remains a critical challenge for real-world deployment: robots with different embodiments—such as bimanual manipulators, humanoid robots, or mobile platforms—require distinct control protocols and safety constraints. The absence of a generic policy that adapts seamlessly across such morphologies limits the universality of current VLA approaches.

This work also does not directly investigate the grounding of instructions that are open-ended or ambiguous. Existing VLAs are trained largely on curated datasets that map well-structured commands to specific actions, but this reliance constrains their semantic understanding. Consequently, when faced with vague or out-of-distribution instructions, they often fail to infer reasonable behaviors. Addressing this limitation will require integrating richer language-understanding modules and more diverse training data to improve robustness in handling underspecified or ambiguous input.

Acknowledgment

This work is supported by the National Robotics Programme (Award #M23NKBK0091, #M25N4N2009). This work is also supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-005), NTU SUG project #025628-00001:Post-training to Improve Embodied AI Agents, and A*STAR SERC CRF funding to Cheston Tan. We thank Wei Qi Toh and Ray Prithviraj for their assistance with robot experiments, and Shaohua Li for manuscript polishing.

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, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, and 5 others. 2024. π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, and 32 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, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*.

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, and 275 others. 2024. Open x-embodiment: Robotic learning datasets and rt-x models. In *ICRA*.

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*.

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, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. 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, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, and 17 others. 2025. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 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, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. 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, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. 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.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. 2024a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. 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.

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

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2025. SAM 2: Segment anything in images and videos. In *ICLR*.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. 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, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. 2024. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv:2412.11974*.

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

Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, and Feifei Feng. 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, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. 2024. Binding touch to everything: Learning unified multimodal tactile representations. In *CVPR*, pages 26340–26353.

Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward H. Adelson. 2024. Transferable tactile transformers for representation learning across diverse sensors and tasks. In *CoRL*.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. 2025. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *ICLR*.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R Sanketi, Grecia Salazar, and 35 others. 2023. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*.

Appendix

A Benchmark Dataset

A.1 Cluttered Tabletop Manipulation Dataset

To evaluate the finetuning behavior of various VLA models under distribution shift, we constructed a custom cluttered tabletop environment using a UR5 robotic arm with a wrist-mounted RealSense RGB-D camera. This setup differs from all existing configurations in the Open-X-Embodiment dataset. Demonstrations for a screwdriver-picking task—amid distractor objects—were collected via teleoperation using a SpaceMouse device. In total, we gathered 163 demonstration episodes[3]. Each episode began with a randomized initial robot pose, followed by an attempt to grasp the target screwdriver.

A.2 Complex Instruction Grounding Dataset

We curated an evaluation dataset for the *complex instruction grounding task* in cluttered scenes[4]. Thirty images were sampled from the action sequences and subsequently categorized based on the number of objects: EASY (<15), MEDIUM (15∼20), and HARD (>20). Objects in the visual scenes were manually annotated using visual prompts and paired with various instructions. The spatial relationship words were illustrate in Table 4.

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

For the complex instruction grounding task, including an annotated visual prompt paired with complex instructions in three forms: implicit, explicit with attributes and spatial references. Additionally, the dataset includes multi-turn questions that refer to more than one object, enabling foundation models to ask clarifying questions to identify the correct object.

[3] https://huggingface.co/datasets/bittdy/pick_screw
[4] https://github.com/xiuchao/InstructionGrounding

Figure 7: Example of visual prompts

- **Implicit Instructions.** 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.

- **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. In Fig. 7, 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.

- **Explicit 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.

- **Multi-Referent Instructions.** This category includes instructions that correspond to multiple valid objects in the scene. For example, in Fig. 7, an instruction like "give me a mug" may refer to several similar items. In such cases, we annotate the data with all candidate object indices, e.g., `[2, 7, 18]`, indicating the set of plausible referents.

Model	Easy			Medium			Hard		
	im	attr	rel	im	attr	rel	im	attr	rel
VLM-LLM	0.050	0.516	0.131	0.010	0.336	0.186	0.000	0.318	0.174
Gemini-2.5-Pro	0.778	0.889	0.830	0.847	0.814	0.815	0.985	0.784	0.858
Gemini-2.0	0.833	0.889	0.774	0.819	0.721	0.642	1.000	0.668	0.469
GPT-5-auto	0.950	1.000	0.867	0.903	0.960	0.899	0.917	0.916	0.816
GPT-5-mini	0.850	1.000	0.894	0.819	0.956	0.823	0.958	0.759	0.813
GPT-4.5	0.944	0.889	0.894	0.917	0.838	0.722	0.958	0.719	0.698
GPT-4o	0.850	1.000	0.778	0.819	0.948	0.680	0.901	0.697	0.469
o4	0.950	1.000	0.907	0.847	0.988	0.837	0.917	0.809	0.804
4o-mini	0.750	0.717	0.550	0.764	0.771	0.596	0.750	0.382	0.248
GPT-4V	0.650	0.750	0.598	0.750	0.737	0.662	0.625	0.417	0.455
Qwen2-VL	0.800	0.917	0.830	0.792	0.756	0.738	0.875	0.700	0.529
LLaMA 3.2 Vision 90B	0.750	0.850	0.704	0.708	0.853	0.711	0.875	0.491	0.521
LLaMA 3.2 Vision 90B-Q4	0.800	0.667	0.598	0.625	0.719	0.554	0.542	0.464	0.300
LLaMA 3.2 Vision 11B	0.650	0.667	0.631	0.764	0.710	0.556	0.833	0.536	0.342
LLaMA 3.2 Vision 11B-Q4	0.650	0.567	0.502	0.694	0.757	0.555	0.542	0.498	0.450

Table 5: Performance on the complex instruction grounding task. Abbreviations: *im* denotes implicit instruction, *attr* denotes attribute-based instruction, and *rel* denotes relation-based instruction.

A human-in-the-loop process was employed to ensure high-quality data collection.

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

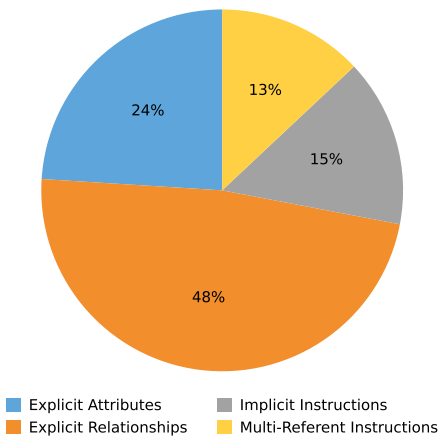


Figure 8: Dataset breakdown by Instruction Types.

This high-quality dataset consisting of 473 instructions, with a detailed breakdown of each instruction type presented in Fig. 8.

B Grounding Experiments

B.1 Complex Instruction Grounding for Goal Specification

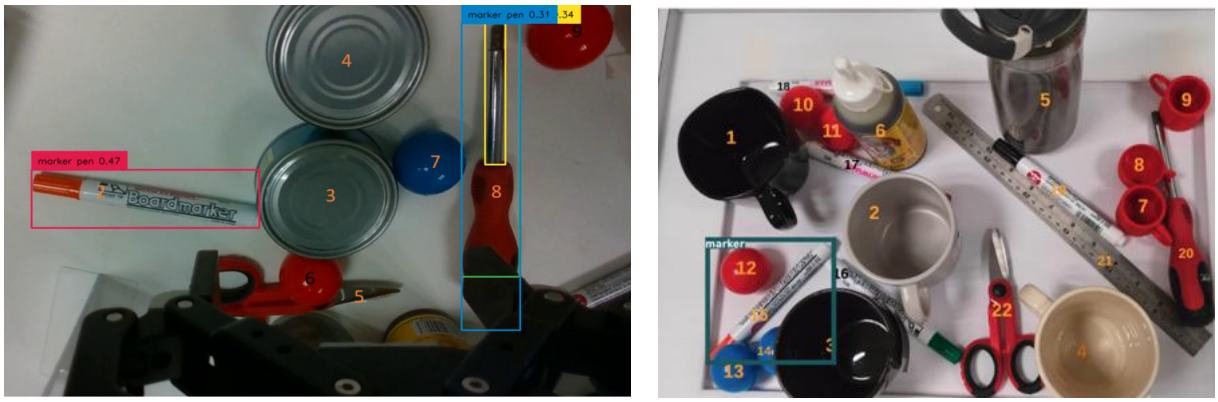
Cross-modal Disambiguation represents a particularly challenging component of goal specification. To quantify the model capability in this dimension, we employed attribute-based and relative relationship instructions to uniquely identify a target among multiple candidates. The goal specification task is formulated as follows:

Given a visual input $I \in \mathbb{R}^{H\times W\times 3}$ and an instruction t, the objective is to predict the target object according to $o^* = \arg\max_{o\in\mathcal{O}} P(o \mid I, t; \theta_{FM})$, where $o^\star \in \mathcal{O}$ and $\mathcal{O}$ denotes the set of candidate objects. Models are evaluated using macro-average accuracy metric.

B.2 Failure Cases of Specialist VLM Pipeline

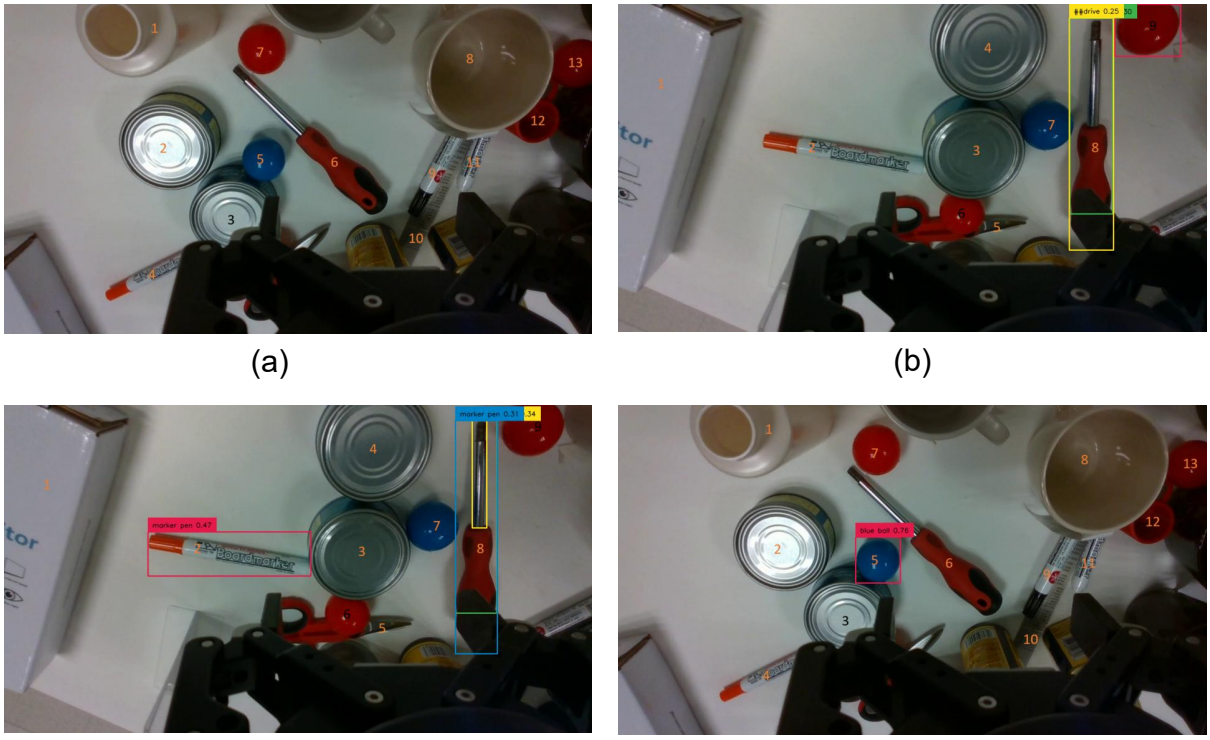
Grounding DINO, despite popular for zero-shot detection, is not robust in open scenes. It successfully detected "blue ball" while failed to detect "ball", indicating its reliance on visual features. Similarly, featureless metal cans pose a great challenge for Grounding DINO, which were almost omitted 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



(a) (b)

Figure 9: Examples of Instruction Grounding. (a) “the marker on the left”, (b) “the marker aligned with the ruler”.



(a) (b)

(c) (d)

Figure 10: Examples of Object Grounding. (a) “ball”, (b) “screwdriver”, (c) “marker pens”, (d) “blue ball”.

bounding boxes. The failure cases were illustrated in the Fig. 9 and Fig. 10.

B.3 Multimodal LLMs Performance

The performance of Multimodal LLMs on complex grounding across EASY, MEDIUM and HARD groups are shown in Table 5.

C Manipulation Experiments

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.

C.1 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. 11. Success rates are summarized in Table 6, demonstrating NORA's superior policy generation capabilities across three task categories: out-of-distribution object grasping, spatial reasoning, and multi-object manipulation.

C.2 Impact of Action Chunking

C.2.1 Action Chunking Performs 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 but the model performance is still degraded.

C.2.2 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 simulation environment and may not reflect real-world performance at high control frequencies.

C.3 Robustness to Disturbance

To evaluate robustness, we selected three straightforward tasks (shown in Fig. 12) 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 8, while the detailed number of successful executions out of 10 trials is summarized in Table 7.

D Modular Claw Machine Prototype

To facilitate the evaluation of different VLMs in robotic manipulation, we developed a voice-controlled testbed using a UR5 robotic arm[5]. The system architecture, shown in Fig. 13, comprises the following five modules:

[5] https://github.com/HRItdy/claw_machine

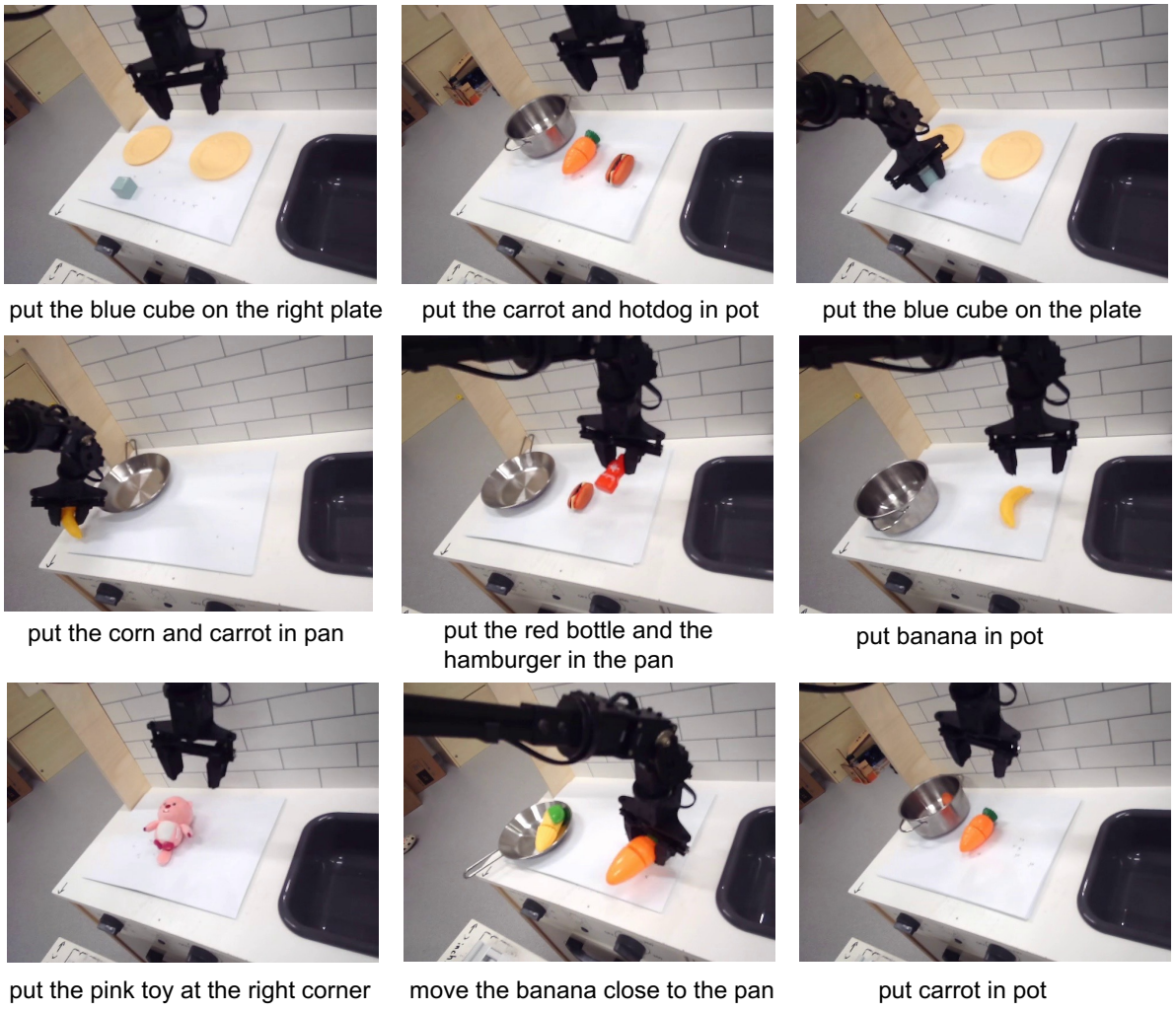


Figure 11: Real-world robot environments and task setups. We evaluate these models across 9 diverse tasks to assess its instruction understanding, spatial reasoning, and multi-task motion planning capabilities.

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 6: Task performance comparison across different categories and models.

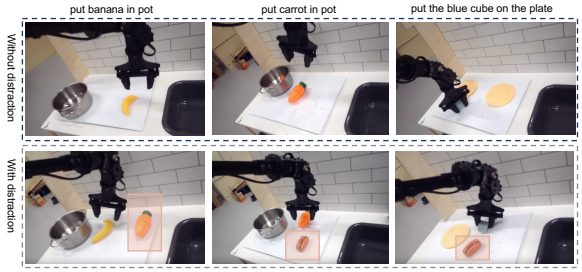


Figure 12: Comparison of tasks with and without distraction.

- **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.

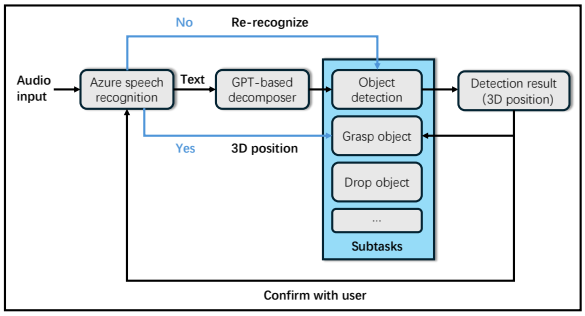


Figure 13: The system architecture of the testbed for VLMs.

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 7: 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.

Table 8: 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

- **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.