StarVLA-α: Reducing Complexity in Vision-Language-Action Systems

Jinhui Ye[1*], Ning Gao[2*], Senqiao Yang[3], Jinliang Zheng[4], Zixuan Wang[1], Yuxin Chen[1], Pengguang Chen[6], Yilun Chen[5†], Shu Liu[6], and Jiaya Jia[1,6†]

[1]The Hong Kong University of Science and Technology
[2]Xi'an Jiaotong University [3]The Chinese University of Hong Kong
[4]Tsinghua University [5]Tongyi Lab, Alibaba Group [6]SmartMore Ltd.

Abstract. Vision-Language-Action (VLA) models have emerged as a promising paradigm for building general-purpose robotic agents. However, the VLA landscape remains highly fragmented and complex: as existing approaches vary substantially in architectures, training data, embodiment configurations, and benchmark-specific engineering. In this work, we introduce **StarVLA-α**, a simple yet strong baseline designed to study VLA design choices under controlled conditions. StarVLA-α deliberately minimizes architectural and pipeline complexity to reduce experimental confounders and enable systematic analysis. Specifically, we re-evaluate several key design axes, including action modeling strategies, robot-specific pretraining, and interface engineering. Across unified multi-benchmark training on LIBERO, SimplerEnv, RoboTwin, and RoboCasa, the same simple baseline remains highly competitive, indicating that a strong VLM backbone combined with minimal design is already sufficient to achieve strong performance without relying on additional architectural complexity or engineering tricks. Notably, our single generalist model outperforms $\pi_{0.5}$ by 20% on the public real-world RoboChallenge benchmark. We expect StarVLA-α to serve as a solid starting point for future research in the VLA regime. Code will be released at `https://github.com/starVLA/starVLA`.

1 Introduction

Recent progress in robotic manipulation has been increasingly driven by Vision-Language-Action (VLA) models, which aim to move beyond task-specific policies toward general-purpose robotic agents. Since the introduction of RT-series [3,8,9] as robotic foundation models, the field has rapidly evolved by leveraging large foundation models, scaling robot data [6,25,63] and general multimodal supervision [8, 12, 27, 67, 69] to achieve impressive policy transferability and task coverages [7–9,27,31,47]. As a result, a growing number of VLA systems demonstrate impressive results across a variety of robotic benchmarks [11, 26, 38, 39, 43, 45]. Meanwhile, open-source efforts [5, 10, 28, 31, 42] have broadened accessibility and accelerated experimentation.

* Equal contribution. † Corresponding author.

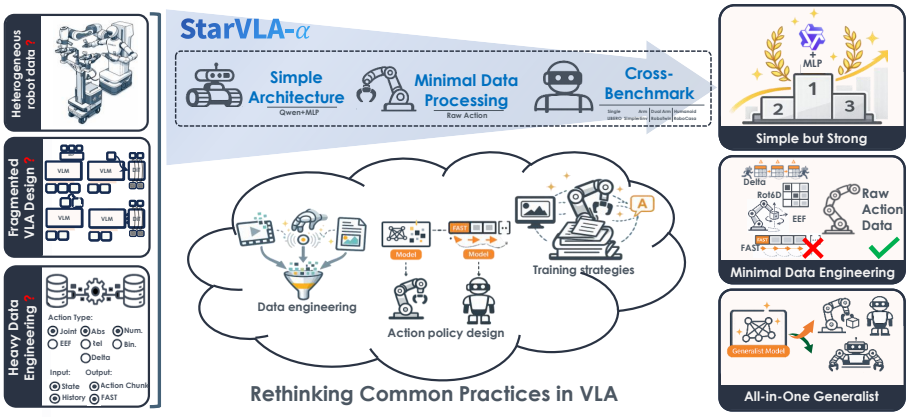

Fig. 1: Current VLA systems are difficult to compare due to heterogeneous robot datasets, fragmented architectures, and heavy benchmark-specific engineering. StarVLA-α removes these confounders with a simple VLM-based architecture, minimal data processing, and unified cross-benchmark training. This controlled baseline enables systematic analysis of action modeling, robot pretraining, and interface design, revealing that many commonly adopted complexities provide limited context-dependent benefits.

Despite the rapid development of VLA systems, the field still lacks a clear understanding of which components actually drive performance gains. Existing systems vary in model architectures, pre-training data, embodiment configurations, and benchmark-specific fine-tuning, making empirical comparison difficult to interpret. Reported improvements are often entangled with dataset choices, preprocessing pipelines, and benchmark-specific engineering, obscuring whether gains arise from modeling innovations or experimental variation. In contrast to vision-language modeling (VLM), where training practices have gradually converged toward standardized recipes [18, 32, 40], VLA research remains highly fragmented. Establishing clearer methodological consensus is therefore increasingly important for guiding future progress in the field.

However, reaching methodological consensus is a challenging and long-standing problem due to substantial heterogeneity across the VLA pipeline as shown in Fig. 1. First, pre-training data and embodiment configurations vary substantially across studies. Rapid evolution of robotic platforms and teleoperation pipelines has led to heterogeneous datasets with incompatible interfaces, action spaces, and normalization schemes [31, 42]. Robot embodiments span single-arm manipulators such as Franka and UR5 [21, 57], wheeled dual-arm systems [1, 22, 23], and humanoid robots [1, 20, 56], accompanied by differences in camera viewpoints and end-effectors, further entangling modeling choices with embodiment-specific preprocessing. Second, modeling and training strategies lack consensus. Existing VLA systems adopt diverse combinations of vision towers, language backbones, and action experts [6, 28, 31, 36, 37, 47, 55], while

design choices such as action parameterization and normalization for continuous robot states and controls remain poorly understood. Third, varied evaluation practices complicate comparison. Benchmark-specific hyperparameter tuning, dataset splits, and action chunking strategies are often required to achieve strong performance [11,33,38,39,45,46], and strong in-benchmark results do not necessarily translate to robustness under broader distribution shifts [24, 46, 50].

To demystify the essential components of VLA systems, we propose **StarVLA-α** upon the infrastructure of StarVLA [16], a simple yet strong baseline that serves as a starting point for systematically studying existing VLA paradigms. It is explicitly designed to reduce experimental confounding and isolate modeling effects. Rather than introducing additional architectural complexity, we deliberately minimize structural variations by employing a pre-trained VLM backbone (Qwen3-VL) without robot-specific pre-training or sophisticated action engineering. We follow official evaluation protocols and avoid benchmark-specific tuning to ensure controlled and reproducible comparisons. The objective is not architectural novelty but methodological clarity: by controlling major sources of variation, StarVLA-α provides a *controlled substrate* for reassessing widely adopted VLA design choices under comparable conditions.

Under this controlled setting, a strong VLM-based baseline matches or exceeds recent VLA systems while keeping the backbone, training data, and training settings identical. Under controlled conditions, we examine the necessity of common VLA design choices along three axes: action head design, robot-specific pretraining, and data/interface engineering. Keeping the backbone, data scale, and training protocol identical, we compare several canonical VLM-to-VLA instantiations within a unified pipeline, including discrete token-based autoregressive decoding (FAST-style), direct continuous action regression with a lightweight MLP head (OpenVLA-OFT-style), diffusion/flow-matching based continuous action generation (π_0-style), and dual-system designs that couple a VLM with a separate low-level action module (GR00T-style), finding that simple MLP action header remains highly competitive while more complex designs provide only scenario-dependent gains (see Sec. 3.1). Robot pretraining by incorporating large-scale action data [15,17] is re-assessed. We observe that heterogeneous pretraining may impair cross-embodiment generalization and that domain-aligned data yields conditional rather than overall improvements (Sec. 3.2). Finally, we revisit common engineering choices (e.g. auxiliary inputs, action output modeling). Overall, removing major confounders reveals that architectural and engineering complexity offers limited and context-dependent gains (Sec. 3.3).

To mitigate potential benchmark-specific bias in single-benchmark evaluation, we further assess robustness under broader generalization regimes. We jointly train a unified model across LIBERO [39], SimplerEnv [44], RoboTwin 2.0 [11], and RoboCasa-GR1 [5,46] without benchmark-specific adaptation, using unified action padding across embodiments (Sec. 4). Under this multi-benchmark setting, the same simple baseline remains competitive, and in several cases superior to task-specific models. These results indicate that strong backbone ini-

tialization and unified training can support cross-task and cross-embodiment generalization without requiring additional architectural complexity.
Our contributions are summarized as follows:

- We present a simple yet strong VLA baseline that removes key confounders, showing that a streamlined VLM design can reach leading performance on four benchmarks spanning five embodiments.
- Under controlled backbone, data, and training settings, we systematically re-evaluate common VLA design choices and find that added architectural/data engineering complexity yields smaller and more context-dependent gains than often assumed.
- We further demonstrate that a single generalist model trained jointly across benchmarks, without task-specific adaptation, can generalize across tasks and embodiments, supported by strong initialization and a standardized pipeline.

2 StarVLA-α

Since the introduction of RT-1 [9] in 2022, Vision–Language–Action (VLA) research has pursued general-purpose embodied agents built on foundation models. Along the way, the community has explored many design dimensions—vision backbones (e.g., SigLIP [72], DINO [48], CLIP [52]), action heads (discrete tokens, continuous regression, diffusion, flow matching), and action/data pipelines (delta vs. relative actions; embodiment-specific preprocessing across eef/joint/6D pose). While these choices have driven steady gains, they have also fragmented the field: systems often become complex, hard to reproduce, and tightly tuned to benchmark-specific details, which can hurt transfer to new embodiments.

Against this backdrop, we ask a simple question: **can we cut through this complexity?** Specifically, we test whether a strong VLM backbone can deliver competitive performance without elaborate architectures or heavy data engineering. To study this, we build a clean, transparent, and robust VLA baseline from scratch (Fig. 2).

2.1 A Simple and Unified VLA Framework

Our framework is guided by a minimal-sufficiency hypothesis: a strong VLM paired with a lightweight action head captures most of the benefits commonly attributed to more complex designs. Here, "clean" refers to two aspects: minimal data processing and a simple architecture.

Minimal data processing. To promote generalization across diverse robot embodiments and benchmarks, we use a single minimal data pipeline shared across all environments. The model takes raw RGB images and the provided language instructions as input, without benchmark-specific engineering or custom formatting. We normalize actions using the training split only (zero mean, unit variance). For evaluation, we follow each benchmark's official protocol. This unified preprocessing makes the framework directly applicable to new robot embodiments and benchmarks without additional adaptation.

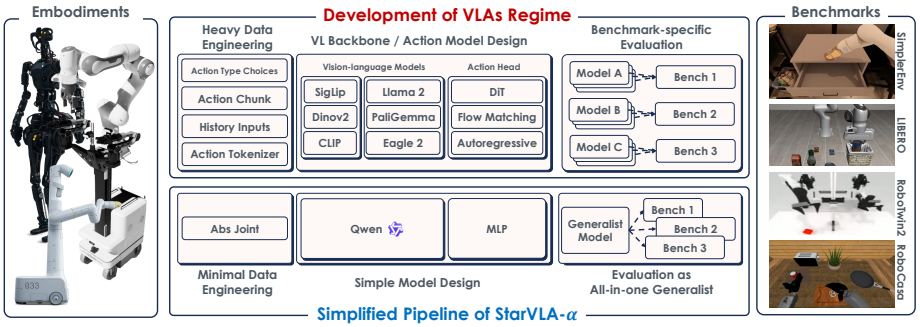

Fig. 2: Overview of StarVLA-α. We use a unified VLM backbone (Qwen3-VL) with minimal preprocessing and a lightweight MLP action head. This simple setup avoids specialized vision encoders, benchmark-specific data pipelines, and complex action heads, while enabling consistent training and evaluation across diverse benchmarks.

Clean architecture. We follow common practice and couple a VLM backbone with a lightweight action head for continuous action prediction. We instantiate the backbone with the Qwen family of models [2,59], specifically Qwen3-VL. We choose Qwen for two reasons: (i) it is a widely adopted open-source VLM with strong community support; and (ii) its unified design natively processes both vision and language inputs, avoiding the need to separately select and combine vision encoders (e.g., CLIP [52], SigLIP [72]). On top of the VLM, we attach a simple MLP action head that reads the hidden state of a designated action token and regresses a chunk of continuous actions. The modular design also allows us to swap in alternative VLM backbones or action heads with minimal changes.

Unified benchmark integration. To enable systematic evaluation, we integrate a diverse suite of manipulation benchmarks (e.g., LIBERO, SimplerEnv, RoboTwin 2.0, and RoboCasa-GR1) into a unified pipeline without benchmark-specific design. For each benchmark, we strictly follow its original data and evaluation protocols, applying only our minimal processing while keeping the action representation consistent. We confine heterogeneity to thin adapters that standardize observation formats, action interfaces, and evaluation entry points. As a result, the same model and training recipe run across all benchmarks without customization, and the framework remains easy to extend to new benchmarks. This setup also enables a more general evaluation regime: **training a single model jointly across all benchmarks**. We refer readers to Sec. 4 for detailed results in this unified multi-benchmark setting.

2.2 Experimental Setup

We evaluate our models on a diverse set of widely used manipulation benchmarks: **LIBERO** [39], **SimplerEnv** [38], the dual-arm benchmark **RoboTwin**

Table 1: Performance comparison of StarVLA-α with existing VLAs. * indicates that both clean and random data are used for training. Default StarVLA-α represents multiple models trained separately on each benchmark-specific dataset, while Generalist represents a single model jointly trained across all datasets.

Method	LIBERO					SimplerEnv			RoboTwin 2.0			RoboCasa-GR1
	Spatial	Object	Goal	Long	avg	WidowX	Google VA	Google VM	clean	clean*	random*	(avg of 24 tasks)
Specialist												
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1	31.3	54.3	63.0	–	–	–	–
π_0	96.8	98.8	95.8	85.2	94.1	27.1	54.8	58.8	46.42	65.9	58.4	–
π_0+FAST	96.4	96.8	88.6	60.2	85.5	39.5	60.5	61.9	–	–	–	–
$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.9	46.9	68.4	72.7	**60.2**	82.7	76.8	37.0
GR00T-N1.6	97.5	98.5	97.5	**94.4**	97.0	62.0	65.3	67.7	–	–	–	47.6
StarVLA-α	**99.0**	**99.8**	98.4	94.0	**98.8**	64.6	**70.2**	**76.0**	50.3	88.2	**88.3**	53.8
StarVLA-α (Generalist)	98.6	99.6	**98.6**	94.2	97.8	**65.2**	69.8	74.3	–	**88.7**	87.8	**57.3**

2.0 [11], and the humanoid benchmark **RoboCasa-GR1** [5, 46]. Benchmark details are provided in the Supplementary Material.

Baselines. We compare StarVLA-α against several representative VLA methods: FAST [49], OpenVLA-OFT [31], π_0 [6], and GR00T-N1.6 [5]. These prior methods are typically trained separately on each benchmark with their own task-specific data processing. For our approach, we consider two training protocols: (1) *Specialist training*, where we train StarVLA-α independently on each benchmark's training set using our unified minimal data pipeline , and (2) *Generalist training*, where we merge all benchmarks' data into a single training set and train a single model. We note that the Generalist model represents a large-scale unified training scenario and is included for completeness, *for direct comparisons under similar computational budgets, we focus primarily on Specialist training.* In the unified setting, actions from different robots are simply padded to a maximum dimension (here 32) with zeros, requiring no per-task engineering; more details are provided in Sec. 4. Training and implementation details are given in the Supplementary Material.

2.3 Main Results

As shown in Table 1, StarVLA-α performs strongly across all benchmarks relative to prior VLA methods. On LIBERO, StarVLA-α achieves an average success rate of 98.8%, outperforming all previous approaches. On SimplerEnv, it exceeds the best existing method by a substantial margin (e.g., +6.8% on Google VM). On the more challenging dual-arm and humanoid settings, StarVLA-α reaches up to 53.8% success, highlighting the strength of a capable VLM backbone even with a lightweight action head.

Moreover, under a unified generalist training setup, we find that training a single model on diverse data achieves competitive per-benchmark performance while notably improving on challenging benchmarks such as RoboCasa-GR1 (Sec. 4).

Together, these results support our central hypothesis: *a strong VLM, paired with a straightforward action head and minimal data preprocessing, can deliver*

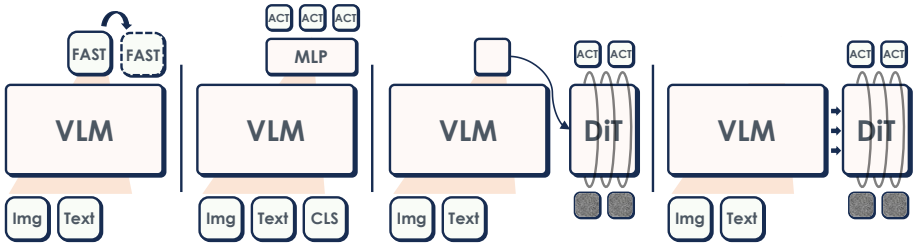

Fig. 3: Action expert designs on StarVLA-α. From left to right: StarVLA-α-FAST, StarVLA-α (MLP regression), StarVLA-α -GR00T (dual-system flow matching), and StarVLA-α-PI (diffusion-style flow matching).

highly competitive performance. More broadly, they suggest a practical way to reduce the field's growing complexity: fix the backbone, standardize the data pipeline, and avoid task-specific engineering. This approach yields a strong, reproducible baseline that can serve as a solid foundation for future work.

3 Rethinking Common Practices in VLA Systems

As described in Sec. 2, StarVLA-α baseline is intentionally simple: it pairs a strong VLM (Qwen3-VL) with a lightweight MLP action head, uses minimally processed data, and introduces no state inputs, history frames, or additional pretraining. Despite this minimal design, StarVLA-α achieves state-of-the-art results across multiple benchmarks and substantially outperforms prior methods. This finding motivates a natural question: **when a strong backbone is available, what actually drives VLA performance?** In this section, we systematically analyze three commonly emphasized design choices: action head architecture, action-specific pretraining, and data engineering.

3.1 Do Different Action Head Designs Matter?

Motivation. Given that our simple MLP head already delivers strong performance, we ask *whether more complex action heads (e.g. fast token predictors, diffusion models, or dual-system designs) provide additional benefits* when paired with the same strong VLM backbone. Prior comparisons have often been confounded by differences in backbones and training recipes; our unified framework enables us to isolate the effect of the action head itself.

Implementation details. As shown in Fig. 3, we evaluate four commonly used designs: (1) **StarVLA-α-FAST**: Discrete action prediction via an autoregressive FAST tokenizer, similar to π_0-FAST [49]. (2) **StarVLA-α**: Continuous action regression with a lightweight MLP head applied to dedicated action tokens, following OpenVLA-OFT [30]. (3) **StarVLA-α-GR00T**: A dual-system architecture where the VLM serves as System 2 (high-level reasoning) and a

Table 2: Performance comparison across action head designs.

Method	LIBERO					SimplerEnv			RoboTwin 2.0			RoboCasa-GR1
	Spatial	Object	Goal	Long	avg	WidowX	Google VA	Google VM	clean	clean*	random*	(avg of 24 tasks)
StarVLA-α	**99.0**	**99.8**	**98.4**	94.0	**98.8**	64.6	70.2	76.0	50.3	**88.2**	88.3	**53.8**
StarVLA-α-FAST	98.2	98.4	97.2	91.6	97.8	35.6	58.8	60.1	46.4	72.5	83.2	45.0
StarVLA-α-GR00T	98.8	99.6	98.4	**95.2**	98.6	65.3	70.7	75.3	48.8	88.0	88.5	52.8
StarVLA-α-π	98.0	99.2	98.2	93.6	98.0	**65.9**	**72.8**	**76.6**	**50.8**	88.1	**88.8**	48.9

Table 3: Effects of additional robotic data pretraining.

Mid-Pretraining	Traj. Num.	RoboTwin-Clean			RoboCasa-GR1		
		Clean 50 × 50	+Random ×100	+Random ×500	24 × 10	24 × 100	24 × 1000
StarVLA-α	-	50.3	78.5	88.2	**9.8**	**39.4**	**53.8**
+ OXE	232.6k	30.2	40.6	83.6	1.2	17.7	27.8
+ InternData-A1	630k	63.6	80.4	88.6	2.8	27.6	35.4
+ RoboTwin-Rand	25k	**79.7**	**84.1**	**88.8**	2.2	27.3	33.3

flow-matching module acts as System 1 for action execution, following GR00T N1.5 [5]. (4) **StarVLA-α-π**: Diffusion-style continuous action prediction using a flow-matching expert, analogous to π_0 [6].

Main results. As shown in Table 2, we compare four action head designs across several settings. Continuous action prediction consistently outperforms discrete action prediction (StarVLA-α-FAST) on nearly all benchmarks. Among the continuous-action variants, however, the three action heads achieve comparable performance. In particular, all methods reach over 98% success on LIBERO and around 65% on WidowX. Notably, the simplest design, StarVLA-α, achieves 53.8% on RoboCasa-GR1.

Takeaway. These results suggest two key observations: (1) continuous action prediction is critical for strong performance and consistently outperforms discrete token-based approaches; and (2) given a powerful VLM, **the choice of continuous action head has limited impact**. Consequently, a lightweight MLP head serves as a simple, efficient, and competitive default. This result indicates that additional architectural complexity in the action head is unnecessary when the underlying VLM backbone is sufficiently strong.

3.2 Does existing action-specific pretraining matter?

Motivation. Most existing VLA models perform large-scale action-specific pretraining before fine-tuning on downstream tasks. For instance, OpenVLA uses the Open X-Embodiment (OXE) dataset [31], $\pi_{0.5}$ leverages diverse robot and multimodal data [28], and GR00T relies on large-scale simulation datasets [5]. Such pretraining is widely regarded as important for strong performance. However, our StarVLA-α baseline, built solely on a pretrained VLM (Qwen3-VL-4B) and without any action-specific data, already achieves competitive results. This observation raises a key question: *given a strong VLM backbone, does additional action-specific pretraining provide further benefits?*

Experimental setups. To answer this question, we use the StarVLA-α architecture and keep all hyperparameters fixed. We compare four pretraining settings and evaluate them on RoboCasa-GR1 and RoboTwin: (1) **StarVLA-α (VLM-based):** No additional pretraining; the pretrained Qwen3-VL model is directly fine-tuned on task-specific data. (2) +**OXE:** The pretrained Qwen3-VL model is first trained on the OXE dataset [15] and then fine-tuned on the task-specific data. (3) +**InternData-A1:** The pretrained Qwen3-VL model is first trained on the InternData-A1 dataset [17], which shares overlapping embodiments (and partially aligned action interfaces) with RoboTwin, and then fine-tuned on the task-specific data. (4) +**RoboTwin-Rand:** Qwen3-VL model is pre-trained on RoboTwin randomized data [11] (within the same domain) and then fine-tuned on the task-specific data.

Results. As shown in Table 3, the StarVLA-α baseline achieves near-best performance when sufficient task-specific data is available, reaching 88.2 and 53.8 on RoboTwin 2.0 and RoboCasa-GR1, respectively. Adding large-scale pretraining data does not consistently improve performance; out-of-domain data such as OXE can even degrade results. Although pretraining with InternData-A1 or RoboTwin data improves RoboTwin performance, particularly in low-data regimes, it still reduces performance on RoboCasa, suggesting that even in-domain gains may not transfer across embodiments or tasks.

Takeaway. Additional action-specific pretraining can improve performance when the pretraining data closely matches the target task, but it may hurt generalization to unseen domains. A strong VLM baseline already provides a solid foundation; further pretraining can therefore act as a double-edged sword and should be applied with caution.

3.3 Is Data Engineering Necessary?

Motivation. Beyond architecture and pretraining, many VLA models rely on various data engineering techniques to improve performance. These include perception-related inputs, such as proprioceptive states and stacked history frames, as well as action representations, such as absolute, delta, or relative actions. While widely used, the necessity of these techniques remains unclear, particularly when a strong VLM backbone is available. In this section, we systematically examine a set of common data engineering choices within StarVLA-α framework. We evaluate them across multiple benchmarks and data scales to determine whether they yield consistent improvements.

Experimental setup. We study four commonly used data engineering choices: (1) **Proprioception [5,6]:** adding robot joint states as input, concatenated with VLM features before the action head. (2) **History frames [34]:** stacking the previous two frames to provide temporal context. (3) **Delta action [19]:** predicting relative changes from the current joint position. (4) **Relative action [19]:** predicting actions in a reference coordinate frame (e.g., end-effector–centric).

Each modification is applied to StarVLA-α while keeping all other training hyperparameters unchanged. We report results on three representative bench-

Table 4: Ablation study on data engineering across benchmarks and data scales.

Mid-Pretraining LIBERO		RoboTwin-2.0			RoboCasa-GR1		
	avg	Clean 50×50	+Random $\times 100$	+Random $\times 500$	24×10	24×100	24×1000
StarVLA-α	**98.8**	50.3	78.5	**88.2**	9.8	39.4	53.8
+ Proprioception	98.4	**60.8**	**79.6**	88.0	12.5	42.1	54.2
+ History frames	97.8	44.8	76.2	87.4	10.2	33.2	52.6
+ Delta action	98.0	48.7	77.8	85.6	**15.8**	**43.2**	54.8
+ Relative action	98.6	51.1	77.9	87.3	13.6	40.6	**55.5**

marks: LIBERO (average over four tasks), RoboTwin 2.0, and RoboCasa-GR1 under different data scales. For RoboTwin 2.0, the data regimes include Clean 50×50, +Random 100, and +Random 500. For RoboCasa, we evaluate with 24×10, 24×100, and 24×1000 demonstrations.

Results. As shown in Table 4, when the dataset is small, for example, Clean 50×50 on RoboTwin 2.0 or 24×10 on RoboCasa-GR1, certain data engineering techniques provide modest improvements. However, once sufficient task-specific data is available, these techniques offer little additional benefit and perform similarly to the baseline without data engineering.

Takeaway. When built upon a strong VLM and a clean codebase, **data engineering techniques can offer modest benefits when task-specific data is limited**. However, their impact **becomes negligible once sufficient task-specific data is available**.

4 All-in-one Evaluation as a Generalist

In Sec. 2, we build a clean VLA framework that achieves strong performance across several individual benchmarks. In Sec. 3, we further rethink several existing techniques and analyze their impact on model training. Hence, after examining the factors that influence training, we move a step further in this section and investigate: *What is an effective evaluation paradigm for assessing whether a model truly possesses generalization ability?*

Existing evaluation patterns. The Embodied AI community shares a unified ambition: to develop a generalist agent that can seamlessly operate across diverse tasks, environments, and robots. In practice, however, the research landscape remains fragmented. Several state-of-the-art systems need to fine-tune their models on benchmark-specific datasets to achieve strong results on individual benchmarks, but their performance drops sharply on others. This leads to a concerning trend in the field: newly proposed policies that excel on one benchmark often suffer sharp performance degradation when transferred to another, making it difficult to demonstrate true generalization ability.

All-in-one evaluation as a generalist. In recent years, large language models (LLMs) have achieved remarkable success, demonstrating generalization capabilities across diverse tasks. A unified evaluation paradigm, which requires a

single model to handle multiple benchmarks simultaneously, has driven progress in generalization within the LLM field. This suggests that an appropriate evaluation paradigm can meaningfully shape both model development and the broader direction of the field. Hence, intuitively, Embodied AI should undergo a similar paradigm shift: evaluating a single model across a wide range of diverse benchmarks to ensure that its capabilities are not tied to any specific environment.

4.1 Task Settings

In this setting, we utilize all datasets to train a single model jointly and directly evaluate it on multiple benchmarks, without any additional fine-tuning on benchmark-specific datasets. Specifically, we select LIBERO, SimplerEnv, RoboTwin 2.0, and RoboCasa-GR1 as the unified benchmark suite and train the model on the combined training sets of these benchmarks.

4.2 Experiments

Implementation details. We set the learning rate as 1×10^{-4}, batchsize as 256 and train on the 5 datasets. In addition, to address the differences in action dimensions across robots, we do not introduce any task-specific design. Instead, we pad the action space of robots with lower degrees of freedom so that all action vectors are uniformly expanded to 32 dimensions in our setting.

Baselines. To further demonstrate the effectiveness of our method and the proposed setting, we report both specialist results, where models are trained only on individual datasets, and results from the generalist training setting. In addition to comparing with our model, we also evaluate several state-of-the-art methods, such as $\pi_{0.5}$ and GR00T-N1.6.

Results. As shown in Table 5, we compare our model trained under the generalist setting, where all datasets are jointly used for training, with specialist models trained on individual benchmarks. Our generalist model consistently achieves sota or competitive performance across most benchmarks. In particular, on the challenging RoboCasa-GR1 benchmark with 24 sub-tasks, our jointly trained model improves performance by 3.5%. These results suggest that a single model can effectively handle diverse tasks and robot embodiments, supporting the development of more unified evaluation paradigms for embodied AI.

4.3 Discussion and Analysis

Our method is simple: it directly pads all actions to the same dimension and uses Qwen3-VL as the pretrained model, yet achieves strong performance. Therefore, in this section, we discuss and analyze *what the most critical factor is in this generalist setting* and *why such a simple method performs so strongly*. We examine this question from multiple perspectives, including action processing, model size, model initialization, and the impact of batch size.

Table 5: Performance comparison between generalist and specialist settings. Specialist represents multiple models trained separately on each benchmark-specific dataset, while Generalist represents a single model jointly trained across all datasets.

Settings	Method	LIBERO					SimplerEnv			RoboTwin 2.0			RoboCasa-GR1
		Spatial	Object	Goal	Long	avg	WidowX	Google VA	Google VM	clean	clean*	random*	(avg of 24 tasks)
Specialist	$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.9	46.9	68.4	72.7	**60.2**	82.7	76.8	37.0
	GR00T-N1.6	97.5	98.5	97.5	94.4	94.1	**67.8**	41.5	35.2	–	–	–	47.6
	StarVLA-α-π	98.0	99.2	98.2	93.6	98.0	65.9	**72.8**	**76.6**	50.8	88.1	**88.8**	48.9
	StarVLA-α-GR00T	98.8	99.6	98.4	**95.2**	98.6	65.3	70.7	75.3	48.8	88.0	88.5	52.8
	StarVLA-α	**99.0**	**99.8**	98.4	94.0	**98.8**	64.6	70.2	76.0	53.4	88.2	88.3	53.8
Generalist	StarVLA-α	98.6	99.6	**98.6**	94.2	97.8	65.2	69.8	74.3	–	**88.7**	87.8	**57.3**

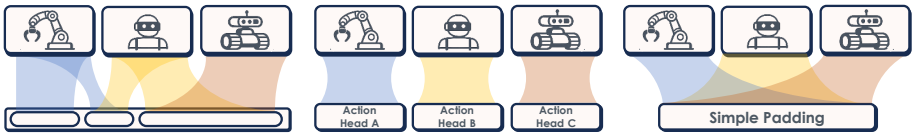

Fig. 4: Comparison of action parameterization for multiple embodiments. Left: RDT Action. Middle: Multi-Action Head. Right: Simple Padding strategy.

Do we truly need specific action designs for each embodiment? Previous studies, such as ABot-VLA [68] and LingBot-VLA [64], have proposed complex, robot-specific solutions, including unified action spaces and multi-action heads tailored to each robotic embodiment. However, modern vision–language models (VLMs) possess sufficient intelligence and parameter capacity to handle diverse tasks. Therefore, can we instead adopt a simple padding strategy and allow the VLA model itself to recognize and manage tasks across multiple embodiments? As shown in Table 6, we compare simple padding strategy with RDT Action and the Multi-Action Head (Fig. 4). Our approach achieves comparable performance on LIBERO and RoboTwin 2.0, while improving results on Google Robot VM and RoboCasa-GR1 by 2.9% and 4.8%, respectively. These results suggest that complex specialist designs may be unnecessary for challenging cross-embodiment tasks.

What is the influence of model size? To further investigate the impact of model size on VLA performance in this general all-in-one setting, we evaluate three pretrained Qwen3-VL models (2B, 4B, and 8B) under the same experimental setup. As shown in Fig. 5, the 4B model achieves significant performance improvements on Simpler compared with the 2B model. Additional results in the Supplementary Material further show that this improvement generalizes beyond a single benchmark, yielding gains of 18.1% on WidowX and 6.6% on RoboCasa-GR1. However, compared with the 4B model, the 8B model does not demonstrate substantial additional improvements, with gains remaining within 1%. These results suggest that, under the current training scale and scenarios, the model size should not be too small, but a 4B parameter scale is sufficient.

Does batchsize matter in the generalist settings? Definitely Yes! As shown in Fig. 5, with additional results provided in the Supplementary Material,

Table 6: Performance comparison of multi-embodiment action parameterization.

Method	LIBERO	SimplerEnv			RoboTwin 2.0		RoboCasa-GR1
	Avg	WidowX	Google VA	Google VM	clean	random	avg
RDT action [42]	97.2	63.9	68.2	71.4	87.2	86.6	52.3
Multi Action Header [58]	97.2	60.6	66.3	67.8	85.6	86.1	53.5
Simple Padding [6]	**97.8**	**65.2**	**69.8**	**74.3**	**88.7**	**87.8**	**57.3**

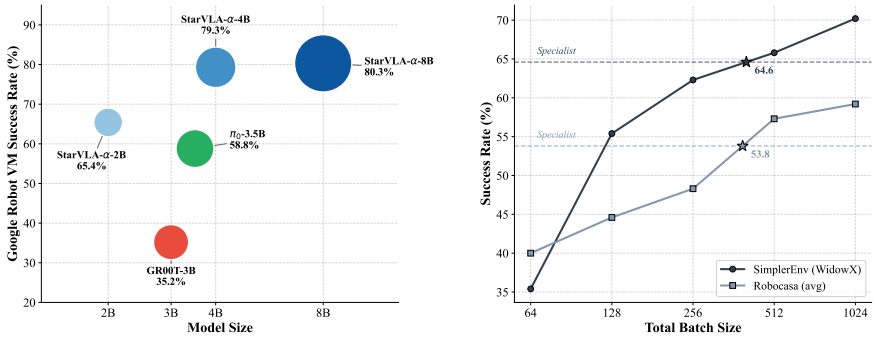

Fig. 5: Scaling trends in VLA training. Left: performance as a function of model size. Right: performance as a function of total batch size.

we evaluate batch sizes of 64, 128, 256, 512, and 1024 under the same total training scale. Performance improves consistently as the batch size increases. With a batch size of 512, the model already achieves strong performance, e.g., 57.3 on RoboCasa-GR1 and 57.2 on RoboTwin-Clean. These results indicate that a larger batch size, which ensures sufficient diversity during training, is a key factor. It helps prevent the model from becoming trapped in local minima, thereby enabling better generalization.

5 Real-World Experiments

In this section, we validate that our minimalist framework remains competitive in physical robot experiments. We evaluate our model on the public real-world benchmark **RoboChallenge** [66], which provides standardized tasks for direct comparison with existing models. Additional real-world OOD experiments are provided in the Appendix.

Benchmark. RoboChallenge is a large-scale real-robot evaluation platform designed to assess learned robotic control policies on physical hardware in a standardized and reproducible manner. We evaluate on the RoboChallenge suite, which contains several tabletop manipulation tasks (e.g., object reorientation, insertion, and multi-stage operations). Following the benchmark protocol, each task is executed multiple times to reduce stochasticity in real-robot trials, and

Table 7: Real-world evaluation as a Generalist in RoboChallenge. SR represents success rate, and score represents progress score.

Robot	Task	StarVLA-α		$\pi_{0.5}$		π_0	
		SR	**score**	**SR**	**score**	**SR**	**score**
ARX5	arrange flowers	40.0	66.5	0.0	30.5	0.0	13.5
	arrange paper cups	20.0	63.0	0.0	31.0	0.0	15.0
	fold dishcloth	0.0	3.5	0.0	0.0	0.0	0.0
	open the drawer	20.0	60.0	50.0	80.0	0.0	20.0
	place shoes on rack	50.0	70.0	0.0	20.0	0.0	16.5
	put cup on coaster	100.0	98.0	70.0	63.0	0.0	0.0
	search green boxes	60.0	58.5	0.0	3.0	0.0	0.0
	sort electronic products	20.0	39.4	0.0	22.5	0.0	22.5
	turn on light switch	50.0	59.0	10.0	25.0	20.0	29.0
	water potted plant	10.0	32.0	0.0	0.0	0.0	0.0
	wipe the table	0.0	44.5	10.0	28.0	0.0	29.0
	Avg.	**33.6**	**54.5**	**12.7**	**27.6**	**3.6**	**14.7**

performance is measured by the average success rate under predefined task success criteria. Additional implementation details are provided in the Appendix.

Results. As shown in Table 7, on the ARX5 robot, we execute the standard set of 11 tasks. The results show that our StarVLA-α achieves a success rate of 33.6 and a progress score of 54.5, which significantly surpass the 12.7 and 27.6 achieved by $\pi_{0.5}$, respectively. These results demonstrate the effectiveness of our StarVLA-α in real-world settings.

6 Related Works

Vision-language-action (VLA) models. The rapid progress of Large Vision-Language Models (VLMs) [4, 41, 59], together with advances in long-form video understanding and agentic world modeling [14, 70], has catalyzed a paradigm shift toward end-to-end Vision-Language-Action (VLA) policies [8, 9]. Building on this foundation, recent work has rapidly expanded the VLA paradigm, exploring diverse architectural designs, including decoupled vision encoder–LLM pipelines [31, 41], native multimodal models [2], and specialized action decoding mechanisms [5, 28]. Recent efforts further study spatial guidance, future visuomotor prediction, visual prompting interfaces, and unified VLA modeling across tasks and embodiments [12, 60, 62, 65]. Meanwhile, training strategies vary substantially across works, spanning robotic datasets [74, 75], human video demonstrations [35, 71], and web data co-training [28]. Alongside a growing number of architectural variants [36, 51, 54] and training recipes [30, 61, 75], these design choices introduce significant heterogeneity across VLA systems, making it difficult to attribute performance improvements to specific algorithmic innovations.

Robotic data engineering and action parameterization. Datasets for robot learning [15, 29] require extensive preprocessing to reconcile differences in control frequencies, camera viewpoints, and action formats [63]. Diverse action parameterization strategies, ranging from discretized token prediction [8, 9, 31] and continuous autoregressive control [30] to action chunking [73] and diffusion-based policies operating [13, 28], have been extensively explored across different models. In addition, various data processing strategies have been shown to affect downstream performance [61], including normalization schemes [31], proprioceptive state conditioning [53], and cross-embodiment padding mechanisms [42, 47]. The reliance on heterogeneous data pipelines tightly couples algorithmic design with engineering choices, obscuring the true sources of performance gains.

7 Conclusion

We introduced **StarVLA-α**, a simple VLA baseline combining a strong VLM backbone with a lightweight MLP action head and minimal data processing, which achieves strong performance across multiple benchmarks and real-world robotic tasks. Controlled experiments with StarVLA-α show that many complex techniques, i.e., sophisticated action header design, heavy data engineering, or task-specific pretraining, are not strictly necessary for generalist robot development. This simplified design reduces architectural complexity, minimizes data engineering, and provides a reproducible and generalizable framework for future VLA research.

Acknowledgements

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

References

1. AgiBot: Agibot official website. `https://www.agibot.com/` (2025)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Belkhale, S., Ding, T., Xiao, T., Sermanet, P., Vuong, Q., Tompson, J., Chebotar, Y., Dwibedi, D., Sadigh, D.: Rt-h: Action hierarchies using language. arXiv preprint arXiv:2403.01823 (2024)
4. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
5. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)

6. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: $\backslash pi_0$: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
7. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: pi_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
8. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818 (2023)
9. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)
10. Cai, J., Cai, Z., Cao, J., Chen, Y., He, Z., Jiang, L., Li, H., Li, H., Li, Y., Liu, Y., et al.: Internvla-a1: Unifying understanding, generation and action for robotic manipulation. arXiv preprint arXiv:2601.02456 (2026)
11. Chen, D., Zhang, J., Mu, T., Tan, Q., Li, Y., Mao, J., Liu, X., Li, K., Qiao, Y., Xiao, F., Ling, Z., Su, H.: Robotwin 2.0: Towards general robot policies with active data generation. arXiv preprint arXiv:2504.13059 (2025), `https://arxiv.org/abs/2504.13059`
12. Chen, X., Chen, Y., Fu, Y., Gao, N., Jia, J., Jin, W., Li, H., Mu, Y., Pang, J., Qiao, Y., et al.: Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. arXiv preprint arXiv:2510.13778 (2025)
13. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion (2024), `https://arxiv.org/abs/2303.04137`
14. Chu, M., Zhang, X.B., Lin, K.Q., Kong, L., Zhang, J., Tu, T., Ma, W., Huang, Z., Yang, S., Huang, W., et al.: Agentic world modeling: Foundations, capabilities, laws, and beyond. arXiv preprint arXiv:2604.22748 (2026)
15. Collaboration, O.X.E., O'Neill, A., Rehman, A., Gupta, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., Tung, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Gupta, A., Wang, A., Kolobov, A., Singh, A., Garg, A., Kembhavi, A., Xie, A., Brohan, A., Raffin, A., Sharma, A., Yavary, A., Jain, A., Balakrishna, A., Wahid, A., Burgess-Limerick, B., Kim, B., Schölkopf, B., Wulfe, B., Ichter, B., Lu, C., Xu, C., Le, C., Finn, C., Wang, C., Xu, C., Chi, C., Huang, C., Chan, C., Agia, C., Pan, C., Fu, C., Devin, C., Xu, D., Morton, D., Driess, D., Chen, D., Pathak, D., Shah, D., Büchler, D., Jayaraman, D., Kalashnikov, D., Sadigh, D., Johns, E., Foster, E., Liu, F., Ceola, F., Xia, F., Zhao, F., Frujeri, F.V., Stulp, F., Zhou, G., Sukhatme, G.S., Salhotra, G., Yan, G., Feng, G., Schiavi, G., Berseth, G., Kahn, G., Yang, G., Wang, G., Su, H., Fang, H.S., Shi, H., Bao, H., Amor, H.B., Christensen, H.I., Furuta, H., Bharadhwaj, H., Walke, H., Fang, H., Ha, H., Mordatch, I., Radosavovic, I., Leal, I., Liang, J., Abou-Chakra, J., Kim, J., Drake, J., Peters, J., Schneider, J., Hsu, J., Vakil, J., Bohg, J., Bingham, J., Wu, J., Gao, J., Hu, J., Wu, J., Wu, J., Sun, J., Luo, J., Gu, J., Tan, J., Oh, J., Wu, J., Lu, J., Yang, J., Malik, J., Silvério, J., Hejna, J., Booher, J., Tompson, J., Yang, J., Salvador, J., Lim, J.J., Han, J., Wang, K., Rao, K., Pertsch, K., Hausman, K., Go, K., Gopalakrishnan, K., Goldberg, K., Byrne, K., Oslund, K., Kawaharazuka, K., Black, K., Lin, K., Zhang, K., Ehsani, K., Lekkala, K., Ellis, K., Rana, K., Srinivasan, K., Fang, K., Singh, K.P., Zeng, K.H., Hatch, K., Hsu, K., Itti, L., Chen, L.Y., Pinto, L., Fei-Fei, L., Tan, L., Fan,

L.J., Ott, L., Lee, L., Weihs, L., Chen, M., Lepert, M., Memmel, M., Tomizuka, M., Itkina, M., Castro, M.G., Spero, M., Du, M., Ahn, M., Yip, M.C., Zhang, M., Ding, M., Heo, M., Srirama, M.K., Sharma, M., Kim, M.J., Kanazawa, N., Hansen, N., Heess, N., Joshi, N.J., Suenderhauf, N., Liu, N., Palo, N.D., Shafiullah, N.M.M., Mees, O., Kroemer, O., Bastani, O., Sanketi, P.R., Miller, P.T., Yin, P., Wohlhart, P., Xu, P., Fagan, P.D., Mitrano, P., Sermanet, P., Abbeel, P., Sundaresan, P., Chen, Q., Vuong, Q., Rafailov, R., Tian, R., Doshi, R., Mart'in-Mart'in, R., Baijal, R., Scalise, R., Hendrix, R., Lin, R., Qian, R., Zhang, R., Mendonca, R., Shah, R., Hoque, R., Julian, R., Bustamante, S., Kirmani, S., Levine, S., Lin, S., Moore, S., Bahl, S., Dass, S., Sonawani, S., Tulsiani, S., Song, S., Xu, S., Haldar, S., Karamcheti, S., Adebola, S., Guist, S., Nasiriany, S., Schaal, S., Welker, S., Tian, S., Ramamoorthy, S., Dasari, S., Belkhale, S., Park, S., Nair, S., Mirchandani, S., Osa, T., Gupta, T., Harada, T., Matsushima, T., Xiao, T., Kollar, T., Yu, T., Ding, T., Davchev, T., Zhao, T.Z., Armstrong, T., Darrell, T., Chung, T., Jain, V., Kumar, V., Vanhoucke, V., Zhan, W., Zhou, W., Burgard, W., Chen, X., Chen, X., Wang, X., Zhu, X., Geng, X., Liu, X., Liangwei, X., Li, X., Pang, Y., Lu, Y., Ma, Y.J., Kim, Y., Chebotar, Y., Zhou, Y., Zhu, Y., Wu, Y., Xu, Y., Wang, Y., Bisk, Y., Dou, Y., Cho, Y., Lee, Y., Cui, Y., Cao, Y., Wu, Y.H., Tang, Y., Zhu, Y., Zhang, Y., Jiang, Y., Li, Y., Li, Y., Iwasawa, Y., Matsuo, Y., Ma, Z., Xu, Z., Cui, Z.J., Zhang, Z., Fu, Z., Lin, Z.: Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864 (2023)
16. Community, S.: Starvla: A lego-like codebase for vision-language-action model developing (2026), https://arxiv.org/abs/2604.05014
17. contributors, I.A.: Interndata-a1. https://github.com/InternRobotics/InternManip (2025)
18. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 36, pp. 49250–49267 (2023)
19. Feng, Y., Zheng, J., Wang, Z., Liu, D., Li, J., Pang, J., Wang, T., Zhan, X.: Demystifying action space design for robotic manipulation policies. arXiv preprint arXiv:2602.23408 (2026)
20. Fourier Intelligence: Fourier gr-1. https://www.fftai.com/products-gr1 (2025)
21. Franka Emika: Franka research 3. https://franka.de/franka-research-3 (2025)
22. Galaxea: Galaxea official website. https://galaxea-ai.com/cn/about (2025)
23. Galbot: Galbot official website. https://www.galbot.com/ (2025)
24. Gao, N., Chen, Y., Yang, S., Chen, X., Tian, Y., Li, H., Huang, H., Wang, H., Wang, T., Pang, J.: Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025), https://arxiv.org/abs/2506.10966
25. Generalist AI: Gen-0: Embodied foundation models that scale with physical interaction. https://generalistai.com/blog/nov-04-2025-GEN-0 (Nov 2025), generalist AI Blog
26. Gu, J., Xiang, F., Li, X., Ling, Z., Liu, X., Mu, T., Tang, Y., Tao, S., Wei, X., Yao, Y., Yuan, X., Xie, P., Huang, Z., Chen, R., Su, H.: Maniskill2: A unified benchmark for generalizable manipulation skills. In: International Conference on Learning Representations (2023)
27. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: Pi0.5: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)

28. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
29. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., et al.: Droid: A large-scale in-the-wild robot manipulation dataset. arXiv preprint arXiv:2403.12945 (2024)
30. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
31. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
32. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
33. Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al.: Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In: Conference on Robot Learning. pp. 80–93. PMLR (2023)
34. Li, H., Yang, S., Chen, Y., Tian, Y., Yang, X., Chen, X., Wang, H., Wang, T., Zhao, F., Lin, D., et al.: Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. arXiv preprint arXiv:2506.19816 (2025)
35. Li, J., Zheng, J., Zheng, Y., Mao, L., Hu, X., Cheng, S., Niu, H., Liu, J., Liu, Y., Liu, J., et al.: Decisionnce: embodied multimodal representations via implicit preference learning. In: Proceedings of the 41st International Conference on Machine Learning. pp. 29461–29488 (2024)
36. Li, Q., Liang, Y., Wang, Z., Luo, L., Chen, X., Liao, M., Wei, F., Deng, Y., Xu, S., Zhang, Y., et al.: Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650 (2024)
37. Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., Li, H., Kong, T.: Vision-language foundation models as effective robot imitators. arXiv preprint arXiv:2311.01378 (2023)
38. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al.: Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941 (2024)
39. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36** (2024)
40. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (2024)
41. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
42. Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., Zhu, J.: Rdt-1b: a diffusion foundation model for bimanual manipulation. arXiv preprint arXiv:2410.07864 (2024)
43. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters **7**(3), 7327–7334 (2022)
44. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple

open-vocabulary object detection. In: European Conference on Computer Vision. pp. 728–755. Springer (2022)
45. Mu, T., Mao, J., Tan, Q., Chen, D., Li, Y., Li, K., Huang, Z., Xie, P., Liu, X., Liu, X., Wang, H., Liu, X., Ling, Z., Tao, S., Jiang, F., Xu, H., Su, H.: Robotwin 1.0: Sim-to-real robot benchmarks are solvable by pre-trained large models as generalist policies. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 32851–32863 (June 2025), https://openaccess.thecvf.com/content/CVPR2025/html/Mu_RoboTwin_1.0_Sim-to-Real_Robot_Benchmarks_Are_Solvable_by_Pre-Trained_Large_CVPR_2025_paper.html
46. Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. arXiv preprint arXiv:2406.02523 (2024)
47. Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Xu, C., Luo, J., Kreiman, T., Tan, Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., Levine, S.: Octo: An open-source generalist robot policy. In: Proceedings of Robotics: Science and Systems. Delft, Netherlands (2024)
48. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
49. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747 (2025)
50. Pumacay, W., Singh, I., Duan, J., Krishna, R., Thomason, J., Fox, D.: The colosseum: A benchmark for evaluating generalization for robotic manipulation. arXiv preprint arXiv:2402.08191 (2024)
51. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
52. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
53. Reuss, M., Zhou, H., Rühle, M., Yağmurlu, Ö.E., Otto, F., Lioutikov, R.: FLOWER: Democratizing generalist robot policies with efficient vision-language-action flow policies. In: 7th Robot Learning Workshop: Towards Robots with Human-Level Abilities (2025), https://openreview.net/forum?id=ifo8oWSLSq
54. Song, W., Zhou, Z., Zhao, H., Chen, J., Ding, P., Yan, H., Huang, Y., Tang, F., Wang, D., Li, H.: Reconvla: Reconstructive vision-language-action model as effective robot perceiver. arXiv preprint arXiv:2508.10333 (2025)
55. Team, X.S.R.: WALL-OSS: Igniting vlms toward the embodied space. arXiv preprint arXiv:2509.06087 (2025), https://x2robot.cn-wlcb.ufileos.com/wall_oss.pdf, code: https://github.com/X-Square-Robot/wall-x
56. Unitree Robotics: Unitree h1 humanoid robot. https://www.unitree.com/h1/ (2025)
57. Universal Robots: Universal robots cb3 series (ur5). https://www.universal-robots.com/cb3/ (2025)
58. Wang, L., Chen, X., Zhao, J., He, K.: Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. Advances in Neural Information Processing Systems **37**, 124420–124450 (2024)

59. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
60. Wang, Q., Li, M., Guan, J., Ye, J., Xie, S., Liu, Y., Chen, J., Liang, Z., Zhang, J., Hu, X., et al.: Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. arXiv preprint arXiv:2605.30280 (2026)
61. Wang, Y., Ding, P., Li, L., Cui, C., Ge, Z., Tong, X., Song, W., Zhao, H., Zhao, W., Hou, P., Huang, S., Tang, Y., Wang, W., Zhang, R., Liu, J., Wang, D.: Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. arXiv preprint arXiv:2509.09372 (2025)
62. Wang, Z., Chen, Y., Liu, Y., Ye, J., Chen, P., Lu, C., Liu, S., Yu, B., Jia, J.: Vp-vla: Visual prompting as an interface for vision-language-action models. arXiv preprint arXiv:2603.22003 (2026)
63. Wu, K., Hou, C., Liu, J., Che, Z., Ju, X., Yang, Z., Li, M., Zhao, Y., Xu, Z., Yang, G., Fan, S., Wang, X., Liao, F., Zhao, Z., Li, G., Jin, Z., Wang, L., Mao, J., Liu, N., Ren, P., Zhang, Q., Lyu, Y., Liu, M., He, J., Luo, Y., Gao, Z., Li, C., Gu, C., Fu, Y., Wu, D., Wang, X., Chen, S., Wang, Z., An, P., Qian, S., Zhang, S., Tang, J.: Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. arXiv preprint arXiv:2412.13877 (2024)
64. Wu, W., Lu, F., Wang, Y., Yang, S., Liu, S., Wang, F., Zhu, Q., Sun, H., Wang, Y., Ma, S., et al.: A pragmatic vla foundation model. arXiv preprint arXiv:2601.18692 (2026)
65. Xu, X., Li, H., Ye, J., Chen, Y., Zeng, J., Chen, X., Xu, L., Lin, D., Li, W., Pang, J.: Futurevla: Joint visuomotor prediction for vision-language-action model. arXiv preprint arXiv:2603.10712 (2026)
66. Yakefu, A., Xie, B., Xu, C., Zhang, E., Zhou, E., Jia, F., Yang, H., Fan, H., Zhang, H., Peng, H., et al.: Robochallenge: Large-scale real-robot evaluation of embodied policies. arXiv preprint arXiv:2510.17950 (2025)
67. Yang, S., Li, H., Chen, Y., Wang, B., Tian, Y., Wang, T., Wang, H., Zhao, F., Liao, Y., Pang, J.: Instructvla: Vision-language-action instruction tuning from understanding to manipulation. arXiv preprint arXiv:2507.17520 (2025)
68. Yang, Y., Zeng, S., Lin, T., Chang, X., Qi, D., Xiao, J., Liu, H., Chen, R., Chen, Y., Huo, D., et al.: Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. arXiv preprint arXiv:2602.11236 (2026)
69. Ye, J., Wang, F., Gao, N., Yu, J., Zhu, Y., Wang, B., Zhang, J., Jin, W., Fu, Y., Zheng, F., Chen, Y., Pang, J.: St4vla: Spatially guided training for vision-language-action models. In: International Conference on Learning Representations (ICLR) (2026)
70. Ye, J., Wang, Z., Sun, H., Chandrasegaran, K., Durante, Z., Eyzaguirre, C., Bisk, Y., Niebles, J.C., Adeli, E., Fei-Fei, L., et al.: Re-thinking temporal search for long-form video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8579–8591 (2025)
71. Ye, S., Jang, J., Jeon, B., Joo, S., Yang, J., Peng, B., Mandlekar, A., Tan, R., Chao, Y.W., Lin, B.Y., et al.: Latent action pretraining from videos. arXiv preprint arXiv:2410.11758 (2024)
72. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11975–11986 (2023)
73. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705 (2023)

74. Zheng, J., Li, J., Liu, D., Zheng, Y., Wang, Z., Ou, Z., Liu, Y., Liu, J., Zhang, Y.Q., Zhan, X.: Universal actions for enhanced embodied foundation models (2025), https://arxiv.org/abs/2501.10105
75. Zheng, J., Li, J., Wang, Z., Liu, D., Kang, X., Feng, Y., Zheng, Y., Zou, J., Chen, Y., Zeng, J., et al.: X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. arXiv preprint arXiv:2510.10274 (2025)