SimVLA: Attributing Gains in VLA Models Through Controlled Ablation

Yuankai Luo[1,2,3] **Woping Chen**[1] **Tong Liang**[1] **Zhenguo Li**[1]*
[1]Frontier Robotics
[2]State Key Laboratory of Novel Software Technology, Nanjing University
[3]School of Artificial Intelligence, Nanjing University
Project website: https://frontierrobo.github.io/SimVLA/

Abstract

Vision-Language-Action (VLA) models often bundle architectural changes with different pretraining data, backbone scales, and optimization recipes, obscuring what actually drives progress. We introduce **SimVLA**, a deliberately minimal VLA—a standard vision-language backbone with a lightweight continuous-action head—as a controlled testbed for this attribution problem; our contribution is methodological and empirical (a controlled attribution study and a minimal-baseline calibration), not a new technique. Across LIBERO, CALVIN, WidowX, and Google Robot, one-knob-at-a-time ablations show that training dynamics are the largest measured driver, producing performance swings of $\Delta\approx$54–94%, while task configuration and architecture have smaller effects under matched settings. Despite using a small backbone and no additional large-scale robot-data VLA pretraining before benchmark-specific training, SimVLA reaches 98.6% average success on LIBERO, remains competitive across the other benchmarks, and transfers to held-out real-robot scenes. These results suggest that simple, carefully controlled VLA baselines remain underexplored and provide a calibration point for future architectural claims within the VLM-encoder plus lightweight-head family.

1 Introduction

The field of Vision-Language-Action (VLA) learning has advanced rapidly, driven by a wave of architectural innovations. Recent methods have proposed increasingly sophisticated designs, ranging from mechanisms that enrich perception with temporal context, to modules that inject explicit 3D spatial awareness, to high-capacity decoders that model complex action distributions. While these contributions have pushed the boundaries of robot capabilities, they also introduce a significant challenge for the research community: attributing performance gains to specific components. Since architectural changes are frequently introduced alongside confounding variables—such as varying pretraining datasets, differing backbone scales, or ad-hoc optimization schedules—it can be difficult to disentangle the impact of a novel mechanism from the underlying training recipe.

To disentangle these confounders, we introduce **SimVLA**, a deliberately minimal VLA that serves as a *controlled testbed* rather than an architectural novelty. As shown in Fig. 2, SimVLA decouples perception from control: a standard pretrained vision-language backbone produces fused representations, and a lightweight action head maps them to continuous actions. We then standardize training dynamics that are often under-specified across papers, including data shuffling, action-space normalization, and optimization schedules. This modular design enables clean ablations along axes that entangled architectures cannot isolate, such as swapping action heads (flow matching, diffusion, and regression) or VLM backbones without redesigning cross-modal adapters.

*Corresponding author.



Figure 1: **Cross-scene zero-shot real-robot deployment.** We deploy SimVLA without any additional fine-tuning on held-out scenes and evaluate it on multi-stage tasks (e.g., store dolls, arrange eggs, fold clothes) that require both dexterous manipulation and semantic understanding.

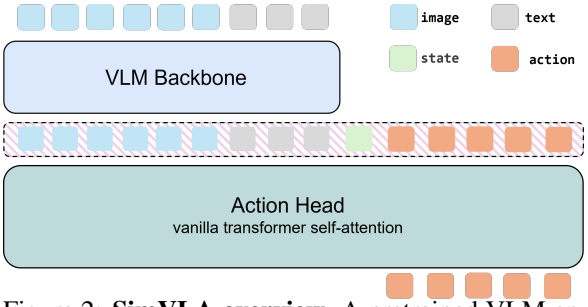


Figure 2: **SimVLA overview.** A pretrained VLM encoder produces fused vision-language tokens once per control step; a lightweight action transformer then outputs a continuous action chunk.

Table 1: **Performance and efficiency summary.** LIBERO success (%) and peak training VRAM at Batch=8 (GB). "B" = backbone scale in billions; SimVLA's 0.5B counts the VLM backbone only.

Model	B	LIBERO Avg (%)	VRAM (GB)
OpenVLA-OFT	7	97.1	62.0
$\pi_{0.5}$	3	96.9	51.3
VLA-Adapter	0.5	97.3	24.7
SimVLA (Ours)	**0.5**	**98.6**	**9.3**

Through systematic cross-benchmark ablations on four datasets (LIBERO, CALVIN, WidowX, Google Robot), we quantify a consistent *attribution hierarchy* in VLA performance: training dynamics have the largest observed effect ($\Delta\approx$54–94%), task configuration ranks second ($\Delta\approx$11–43%), and architecture has the smallest residual effect ($\Delta\approx$0.6–15.6%). This finding parallels well-established observations in language modeling, where data scale, data quality, and optimization recipes can outweigh architectural changes [Hoffmann et al., 2022, Touvron et al., 2023]; however, this attribution hierarchy has not been systematically quantified in the VLA setting. These observations together form the paper's central empirical claim: an attribution hierarchy that future VLA research can use to interpret and control its reported gains.

Despite its simplicity, SimVLA is both effective and efficient. Figure 1 illustrates SimVLA's cross-scene zero-shot generalization capability when deployed on the Galaxea R1 Lite; additional details about the robot platform and training data are provided in Section 4.1. Table 1 further presents a representative example in which our model outperforms multi-billion-parameter baselines while requiring substantially lower peak training VRAM.

Our main contributions are:

- **Methodological and empirical contribution.** We present a controlled attribution study and a minimal-baseline calibration, not novel training techniques. Through strict one-knob-at-a-time ablations on four benchmarks, we identify a **quantitative attribution hierarchy** for VLA performance: training dynamics ($\Delta\approx$54–94%) $>$ task configuration ($\Delta\approx$11–43%) $>$ architecture

($\Delta \approx 0.6$–15.6%). This hierarchy provides a calibration scale for evaluating future architectural claims under matched training and task-configuration controls, within the VLM-encoder plus lightweight-head family.
- We demonstrate that SimVLA—with a small backbone and **no additional large-scale robot-data VLA pretraining before benchmark-specific finetuning**—achieves state-of-the-art or competitive results on LIBERO (98.6%), CALVIN ABC→D (4.47), WidowX (95.8%), and Google Robot (77.0%).

2 Related Work

Recent VLA research has improved robot control through increasingly rich multimodal architectures. We group these advances along three axes and refer interested readers to recent surveys for broader coverage of the field [Xu et al., 2025, Sapkota et al., 2025, Zhang et al., 2025b].

Visual and Temporal Augmentation. Beyond static RGB inputs used by early VLAs such as OpenVLA [Kim et al., 2024], recent methods add explicit motion, prediction, or memory signals. FlowVLA, CoT-VLA, TraceVLA, and 4D-VLA inject optical flow, sub-goal images, trajectory traces, or spatiotemporal features [Zhong et al., 2025, Zhao et al., 2025, Zheng et al., 2024, Zhang et al., 2025a]; PixelVLA adds pixel-level perception via multiscale encoders [Liang et al., 2025a] and ReconVLA adds a reconstruction objective [Song et al., 2025b]; WorldVLA, Dream-VLA, ThinkAct, IntentionVLA, FPC-VLA, MemoryVLA, and HAMLET add future prediction, planning, intention reasoning, failure correction, or history-aware memory [Cen et al., 2025, Zhang et al., 2025c, Huang et al., 2025, Chen et al., 2025, Yang et al., 2025, Shi et al., 2025, Koo et al., 2025]. These additions improve temporal reasoning but often increase computational overhead and deployment complexity.

Geometric and 3D Priors. Another line injects spatial structure that 2D VLM backbones may not represent precisely. 4D-VLA and SpatialVLA add 3D positional encodings [Zhang et al., 2025a, Qu et al., 2025]; GraspVLA and MolmoAct introduce progressive grasp reasoning and visual reasoning traces [Deng et al., 2025, Lee et al., 2025]; GeoVLA, FALCON, and DepthVLA process point clouds, spatial tokens, or predicted depth features [Sun et al., 2025, Zhang et al., 2025d, Yuan et al., 2025]. These designs can improve geometric precision, but they also introduce sensor dependencies or heavy 3D encoders that may reduce embodiment generality.

Action Representations and Architectures. Recent work also studies how actions should be represented and decoded. FAST, PD-VLA, OpenVLA-OFT, VLA-Adapter, FLOWER, X-VLA, NORA, SmolVLA, GR00T-N1, and UnifiedVLA improve efficiency through tokenization, direct regression, adapters, soft prompts, embodiment-specific designs, or unified autoregressive token modeling [Pertsch et al., 2025, Song et al., 2025a, Kim et al., 2025, Wang et al., 2025b, Reuss et al., 2025, Zheng et al., 2025b, Hung et al., 2025, Shukor et al., 2025, NVIDIA et al., 2025, Wang et al., 2025c]. Diffusion and latent-action policies such as Diffusion Policy, π_0, $\pi_{0.5}$, DD-VLA, UniVLA, UniAct, and VQ-VLA improve action expressivity [Chi et al., 2024, Black et al., 2024, 2025, Liang et al., 2025b, Bu et al., 2025, Zheng et al., 2025a, Wang et al., 2025a], sometimes at the cost of latency or training complexity.

Our work offers a counterpoint to this trend. SimVLA asks whether a simple VLM encoder plus lightweight continuous-action head, paired with a carefully controlled training recipe, can match more elaborate systems. Concurrent VLANeXt [Wu et al., 2026] also argues that VLA progress depends on design and training recipes, and progressively builds a strong model on LIBERO/LIBERO-plus by distilling practical design findings. SimVLA is complementary: rather than optimizing a final recipe, we use a deliberately minimal VLM-encoder-plus-action-head testbed and one-knob-at-a-time ablations across four benchmarks to quantify which reported gains remain after training dynamics and task configuration are controlled. We provide a summary mapping of these aforementioned baselines to their respective design axes in Table 8 (Appendix).

3 SimVLA: A Simple VLA Baseline

As introduced in Section 2, recent VLA models often conflate architecture, task configuration, and training recipe changes, making it difficult to identify which choices actually drive performance. We therefore use SimVLA as a deliberately minimal and standardized empirical testbed: a pretrained VLM encoder coupled to a lightweight continuous-action head. This design fixes the architectural

surface enough to support one-knob-at-a-time comparisons, allowing the evaluation in Section 4 to estimate an attribution hierarchy over training dynamics, task configuration, and architecture.

3.1 Preliminaries

Problem Formulation. We model the conditional distribution of a future action chunk $A_t = [a_t, a_{t+1}, \ldots, a_{t+H-1}] \in \mathbb{R}^{H \times d_a}$ given an observation o_t. The observation contains the respective multi-view RGB images $I_t^1, \ldots, I_t^n$, the pairing language instruction ℓ_t, and the robot proprioception (state) s_t: $o_t = [I_t^1, \ldots, I_t^n, \ell_t, s_t]$.

VLM Backbone Encoder. Following the standard late-fusion paradigm [Black et al., 2024], we employ a pretrained VLM E_ϕ to map multi-view RGB observations and the corresponding language instruction into a shared token representation:

$$Z_t = E_\phi(I_t^1, \ldots, I_t^n, \ell_t).$$

We treat the VLM as a *perception-language encoder*, not an action generator: semantic understanding is handled by the VLM, while continuous control is delegated to a lightweight action head. The VLM is not frozen by default; we jointly fine-tune it with the action head, optionally using a short warm-up for stability.

Action Head. SimVLA uses a vanilla Transformer encoder [Vaswani et al., 2017] as a lightweight action head over fused VLM tokens Z_t, proprioception s_t, and action-query tokens. This interface is objective-agnostic: different continuous-action objectives can be instantiated while keeping the VLM backbone, action-head architecture, and training recipe fixed.

Default Action Objective. The reported SimVLA default instantiates this interface with conditional flow matching [Lipman et al., 2022, Black et al., 2024]. Let x denote a normalized action chunk and $\epsilon \sim \mathcal{N}(0, I)$ denote Gaussian noise. We sample $t \in (0, 1]$, construct $x_t = t\epsilon + (1 - t)x$, and train the action head $v_\theta(x_t, o_t, t)$ with:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\|v_\theta(x_t, o_t, t) - (\epsilon - x)\|_2^2\right].$$

At inference time, the default head integrates the learned vector field with a small number of Euler steps to generate smooth action chunks. Under identical training conditions, flow matching outperforms DDPM diffusion and MSE regression in our LIBERO action-objective ablation (98.6% vs. 95.5% and 95.3%; Table 7). The default training and inference settings employed by SimVLA are detailed in the Appendix (Table 9).

3.2 SimVLA Architecture

Design Principle. SimVLA adopts an intentionally minimal design (Figure 2): a vision-language encoder produces fused representations once per control step, and a lightweight action transformer generates continuous action chunks under a configurable action objective.

At each timestep, the fused vision-language tokens Z_t are first obtained from the VLM backbone. We then construct the input sequence to the action head by concatenating projected VLM tokens, a broadcasted proprioceptive state embedding, a timestep embedding, and objective-specific action-query tokens. This unified token sequence is processed by a vanilla Transformer encoder with pure self-attention [Vaswani et al., 2017], without cross-attention, memory modules, or modality-specific routing.

Rationale and Practical Advantages. Self-attention over concatenated tokens provides a neutral integration mechanism: more specialized bridges or conditional normalization may help, but they also add confounders. Executing the VLM once per control step and delegating action generation to the lightweight head also reduces inference cost.

3.3 Controlled Training and Inference Setup

For SimVLA to serve as an attribution testbed, its reference setup must be specified as part of the experimental control rather than treated as incidental implementation detail. We therefore define the action representation, data handling, optimization, architecture configuration, and inference procedure used by the policy; ablations then vary one of these choices while holding the others fixed.

Action Representation and Normalization. The policy predicts normalized continuous action chunks of horizon H and executes them in a receding-horizon manner. Normalization statistics are computed per action dimension from the training set, and proprioceptive states are normalized when applicable. We set $H{=}10$ for LIBERO and $H{=}30$ for CALVIN, WidowX, Google Robot, and real-robot training, with horizon changes treated as controlled task-configuration ablations.

Data Handling. Training uses trajectory shuffling as the reference data-ordering choice. Because demonstration ordering can affect optimization and long-horizon generalization, we treat shuffling as an explicit controlled variable rather than as a hidden data-loader choice.

Optimization Dynamics. For optimization, we fix batch size, training budget, learning-rate schedule, and backbone adaptation settings for each benchmark. In the main LIBERO configuration, this corresponds to AdamW with learning rate 2×10^{-4}, no warm-up, no scheduler, and a VLM learning-rate multiplier of 0.1. Ablations vary these optimization knobs while holding the architecture and task configuration fixed.

Architecture Configuration. We treat action transformer scale, VLM backbone choice, and information injection (concatenation, cross-attention, AdaLN) as controlled architecture knobs evaluated only after data and optimization controls are matched. The reference policy uses the large action head $\{1024, 24, 16\}$ for $\{$hidden size, depth, heads$\}$ ($\sim$300M parameters, $\sim$0.80B trainable parameters in total with the 0.5B backbone); the small head $\{768, 12, 12\}$ has $\sim$80M parameters ($\sim$0.58B in total). Throughout the paper, "B" denotes backbone scale only. Benchmark-specific backbone choices, optimizer settings, and budgets are in the Appendix.

Inference. At inference time, SimVLA follows an *encode-once, predict-in-the-head* workflow. Given an observation, the VLM backbone is executed once to obtain fused tokens, after which the lightweight action head generates a continuous action chunk under the configured objective. The resulting actions are post-processed and executed in a receding-horizon fashion.

4 Evaluation

4.1 Experimental Setup

We follow standard evaluation protocols for each benchmark.

LIBERO [Liu et al., 2023]. We utilize all four standard suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, comprising 10 tasks per suite with 500 expert demonstrations each. This serves as our primary testbed to evaluate the model's long-horizon consistency and generalization performance.

LIBERO-PRO [Zhou et al., 2025]. To address the memorization issue in current VLA studies, we evaluate on LIBERO-PRO, a robust benchmark that introduces systematic perturbations across four dimensions: object appearance (Obj), spatial layout (Pos), language instructions (Sem), and task goals (Task).

SimplerEnv [Li et al., 2024c]. We evaluate on Simpler-Fractal (Google Robot) and Simpler-Bridge (WidowX) to assess the model's performance in high-fidelity simulated environments. For Simpler-Fractal, we report variant aggregation scores to test the policy's robustness against diverse scene variations. For Simpler-Bridge, the evaluation focuses on real-to-sim transfer across tasks.

CALVIN [Mees et al., 2022]. We use the standard ABC$\rightarrow$D setting to evaluate long-horizon language-conditioned manipulation and cross-scene zero-shot generalization to an unseen environment. Following the official protocol, each evaluation episode contains a chain of five language instructions, and we report the success rates of completing at least 1/2/3/4/5 consecutive subtasks, together with the average successful sequence length.

Evaluation Protocol. Following recent works [Zheng et al., 2025b, Liang et al., 2025b, Wang et al., 2025b], we train a *single generalist policy* on the union of all standard LIBERO datasets (Spatial, Object, Goal, Long). For LIBERO-PRO, we directly evaluate this policy without additional fine-tuning to strictly test cross-scene zero-shot robustness. For SimplerEnv, we train our model on the Fractal [Brohan et al., 2023] and BridgeData-V2 datasets [Walke et al., 2024], strictly following recent work [Zheng et al., 2025b]. For CALVIN, we train on the standard ABC split and evaluate cross-scene zero-shot on the held-out D environment under the official multi-step evaluation protocol [Zheng

Table 2: **Comparison on the LIBERO benchmark.** Success rate (%) per suite and overall average. **Bold*** = best; **Bold** = second-best. "B" = backbone scale in billions.

Model	B	Spatial	Object	Goal	Long	Avg
Large Models ($\geq$ 4B)						
UniVLA [Bu et al., 2025]	9	96.5	96.8	95.6	92.0	95.2
FlowVLA [Zhong et al., 2025]	8.5	93.2	95.0	91.6	72.6	88.1
UnifiedVLA [Wang et al., 2025c]	8.5	95.4	98.8	93.6	94.0	95.5
OpenVLA [Kim et al., 2024]	7	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [Kim et al., 2025]	7	97.6	98.4	97.9	94.5	97.1
DD-VLA [Liang et al., 2025b]	7	97.2	98.6	97.4	92.0	96.3
MemoryVLA [Shi et al., 2025]	7	98.4	98.4	96.4	93.4	96.7
PD-VLA [Song et al., 2025a]	7	95.5	96.7	94.9	91.7	94.7
MolmoAct [Lee et al., 2025]	7	87.0	95.4	87.6	77.2	86.6
ThinkAct [Huang et al., 2025]	7	88.3	91.4	87.1	70.9	84.4
CoT-VLA [Zhao et al., 2025]	7	87.5	91.6	87.6	69.0	83.9
WorldVLA [Cen et al., 2025]	7	87.6	96.2	83.4	60.0	81.8
TraceVLA [Zheng et al., 2024]	7	84.6	85.2	75.1	54.1	74.8
FPC-VLA [Yang et al., 2025]	7	87.0	92.0	86.2	82.2	86.9
4D-VLA [Zhang et al., 2025a]	4	88.9	95.2	90.9	79.1	88.6
SpatialVLA [Qu et al., 2025]	4	88.2	89.9	78.6	55.5	78.1
Small Models ($<$ 4B)						
π_0 [Black et al., 2024]	3	96.8	98.8	95.8	85.2	94.2
π_0-FAST [Pertsch et al., 2025]	3	96.4	96.8	88.6	60.2	85.5
$\pi_{0.5}$ [Black et al., 2025]	3	98.8	98.2	**98.0**	92.4	96.9
NORA [Hung et al., 2025]	3	92.2	95.4	89.4	74.6	87.9
SmolVLA [Shukor et al., 2025]	2.2	93.0	94.0	91.0	77.0	88.8
VLANeXt [Wu et al., 2026]	2	**99.0**	99.2	96.6	94.6	97.4
GR00T-N1 [NVIDIA et al., 2025]	2	94.4	97.6	93.0	90.6	93.9
GraspVLA [Deng et al., 2025]	1.8	-	94.1	91.2	82.0	89.1
FLOWER [Reuss et al., 2025]	1	97.5	99.1	96.1	94.9	96.9
Tiny Models ($<$ 1B)						
X-VLA [Zheng et al., 2025b]	0.9	98.2	98.6	97.8	**97.6***	**98.1**
VLA-Adapter [Wang et al., 2025b]	0.5	97.8	**99.2**	97.2	95.0	97.3
VLA-OS [Gao et al., 2025]	0.5	87.0	96.5	92.7	66.0	85.6
UniAct [Zheng et al., 2025a]	0.5	77.0	87.0	77.0	70.0	77.8
SimVLA	0.5	**99.6***	**99.8***	**98.6***	**96.4**	**98.6***

Table 3: **Robustness on LIBERO-PRO.** Success rate (%) across perturbation dimensions: Original (Ori), Object (Obj), Position (Pos), Semantic (Sem), Task (Task). **Bold** = best.

Suite	Method	Ori	Obj	Pos	Sem	Task
Spatial	OpenVLA [Kim et al., 2024]	98.0	97.0	0.0	97.0	0.0
	$\pi_{0.5}$ [Black et al., 2025]	98.0	97.0	20.0	97.0	**1.0**
	SimVLA	**99.0**	**98.0**	**29.0**	**98.0**	0.0
Object	OpenVLA [Kim et al., 2024]	99.0	**98.0**	0.0	98.0	0.0
	$\pi_{0.5}$ [Black et al., 2025]	98.0	**98.0**	**17.0**	96.0	1.0
	SimVLA	**100.0**	85.0	1.0	**100.0**	**4.0**
Goal	OpenVLA [Kim et al., 2024]	98.0	96.0	0.0	98.0	0.0
	$\pi_{0.5}$ [Black et al., 2025]	97.0	**97.0**	**38.0**	97.0	0.0
	SimVLA	**99.0**	82.0	0.0	**99.0**	**10.0**
Long	OpenVLA [Kim et al., 2024]	93.0	81.0	0.0	96.0	0.0
	$\pi_{0.5}$ [Black et al., 2025]	93.0	**92.0**	**8.0**	93.0	1.0
	SimVLA	**96.0**	61.0	3.0	**98.0**	**10.0**

Table 4: **Comparison on CALVIN ABC→D;** success rate (%).

Model	SR1	SR2	SR3	SR4	SR5	Avg.len
Large ($\geq$ 4B)						
UniVLA [Bu et al., 2025]	95.5	85.8	75.4	66.9	56.5	3.80
UnifiedVLA [Wang et al., 2025c]	98.9	94.8	89.0	82.8	75.1	4.41
OpenVLA [Kim et al., 2024]	91.3	77.8	62.0	52.1	43.5	3.27
OpenVLA-OFT [Kim et al., 2025]	96.3	89.1	82.4	75.8	66.5	4.10
HiF-VLA [Lin et al., 2025]	98.5	94.1	88.1	81.4	73.1	4.35
Tiny ($<$ 1B)						
Seer [Tian et al., 2024]	96.3	91.6	86.1	80.3	74.0	4.28
VLA-Adapter [Wang et al., 2025b]	99.1	94.6	88.8	82.8	76.5	4.42
X-VLA [Zheng et al., 2025b]	97.1	92.6	88.5	**84.4**	**78.8**	4.43
SimVLA	**99.4**	**96.4**	**89.8**	83.7	77.2	**4.47**

et al., 2025b]. Across all benchmarks, we train directly from a pretrained VLM backbone without additional large-scale robot-data VLA pretraining before benchmark-specific training.

Critical Hyperparameters. A key finding of this study is that performance is highly sensitive to a small set of training and design choices, which can easily overshadow architectural differences if left under-tuned. To make our comparisons transparent, we summarize the concrete settings we tried:

- **Data & representation.** We vary the *action-chunk horizon* $H \in \{10, 20, 30\}$. We ablate *data shuffling* (shuffle vs. no shuffle). We ablate *normalization* (on/off) for both actions and proprioception; when enabled, we compute per-dimension statistics (mean/std, with robust quantile estimates) on the training set and normalize accordingly.
- **Optimization dynamics.** We sweep the *learning rate* over $\{5\times10^{-5}, 10^{-4}, 2\times10^{-4}, 5\times10^{-4}\}$, *warm-up steps* over $\{0, 1000\}$, and *the scheduler* over {cosine decay, none}. For all variants, we keep the training budget fixed (batch size and total steps/epochs).
- **Architecture configuration.** *Action transformer scale:* we use two configurations (small vs. large), as in our training scripts, e.g., $\{768, 12, 12\}$ ($\sim$80M params) vs. $\{1024, 24, 16\}$ ($\sim$300M params) for {hidden size, depth, # heads}. *VLM backbone:* we compare Florence-2 (0.9B) [Xiao et al., 2024] and SmolVLM-0.5B [Marafioti et al., 2025], and additionally evaluate Qwen3-VL-2B [Qwen Team, 2025] at the default LIBERO configuration (Appendix H.4). *Information injection:* we compare (i) token concatenation with pure self-attention (default), (ii) cross-attention injection, and (iii) conditional AdaLN.

Detailed hyperparameters are provided in the Appendix D.

Baselines. We compare against 34 representative VLA policies using the results reported in their original papers (none are reproduced under our setup). SimVLA numbers in the tables are single seeds (seed 0); three-seed results are in Appendix H.3 (Tables 20–23).

Real-robot (Galaxea R1 Lite). Beyond simulation, we evaluate cross-scene zero-shot generalization on a mobile bimanual robot. We train SimVLA and a $\pi_{0.5}$ baseline on the 500-hour Galaxea Open-

Table 5: **Comparison on WidowX robot tasks;** success rates (%).

Method	Spoon	Carrot	Stack	Eggplant	Avg
Large (≥ 4B)					
RT-1-X [Collaboration et al., 2025]	0.0	4.2	0.0	0.0	1.1
Octo-Base [Team et al., 2024]	12.5	8.3	0.0	43.1	16.0
OpenVLA [Kim et al., 2024]	0.0	0.0	0.0	4.1	1.0
OpenVLA-OFT [Kim et al., 2025]	12.5	4.2	8.3	37.5	15.6
DD-VLA [Liang et al., 2025b]	29.2	29.2	20.8	70.8	37.5
RoboVLM [Li et al., 2024b]	29.2	25.0	12.5	58.3	31.3
SpatialVLA [Qu et al., 2025]	16.7	25.0	29.2	100.0	42.7
MemoryVLA [Shi et al., 2025]	75.0	75.0	37.5	100.0	71.9
ThinkAct [Huang et al., 2025]	58.3	37.5	8.7	70.8	43.8
FPC-VLA [Yang et al., 2025]	79.2	58.3	45.8	75.0	64.6
Small (< 4B)					
Octo-Small [Team et al., 2024]	47.2	9.7	4.2	56.9	29.5
π_0 [Black et al., 2024]	29.1	0.0	16.7	62.5	27.1
π_0-FAST [Pertsch et al., 2025]	29.1	21.9	10.8	66.6	32.1
GR00T-N1 [NVIDIA et al., 2025]	62.5	45.8	16.7	20.8	36.5
FLOWER [Reuss et al., 2025]	71.0	13.0	8.0	88.0	45.0
Tiny (< 1B)					
X-VLA [Zheng et al., 2025b]	100	91.7	95.8	95.8	95.8
SimVLA	**100**	**100**	**91.7**	**91.7**	**95.8**

Table 6: **Comparison on Google Robot tasks;** success rates (%). Variant aggregation.

Model	Pick	Move	Open	Avg
Large (≥ 4B)				
RT-1-X [Collaboration et al., 2025]	49.0	32.3	29.4	36.9
RT-2-X [Collaboration et al., 2025]	82.3	79.2	35.3	65.6
Octo-Base [Team et al., 2024]	0.6	3.1	1.1	1.6
OpenVLA [Kim et al., 2024]	54.5	47.7	17.7	40.0
OpenVLA-OFT [Kim et al., 2025]	65.3	59.0	12.2	45.5
RoboVLM [Li et al., 2024b]	75.6	60.0	10.6	48.7
TraceVLA [Zheng et al., 2024]	60.0	56.4	31.0	49.1
DD-VLA [Liang et al., 2025b]	82.5	64.6	23.6	56.9
SpatialVLA [Qu et al., 2025]	88.0	72.7	41.8	67.5
ThinkAct [Huang et al., 2025]	84.0	63.8	47.6	65.1
Small (< 4B)				
π_0 [Black et al., 2024]	75.2	63.7	25.6	54.8
π_0-FAST [Pertsch et al., 2025]	77.6	68.2	31.3	59.0
GR00T-N1 [NVIDIA et al., 2025]	78.8	62.5	13.2	51.5
Tiny (< 1B)				
X-VLA [Zheng et al., 2025b]	85.5	79.8	61.9	75.7
SimVLA	**82.3**	**81.0**	**67.7**	**77.0**

World Dataset [Jiang et al., 2025], then deploy both in held-out scenes without additional fine-tuning. We evaluate on eight multi-stage manipulation tasks that emphasize dexterous, fine-grained manipulation under diverse scenes and objects: *store the dolls*, *arrange eggs*, *put the flowers in the vase*, *put the pen into the pen holder*, *wipe the desktop*, *fold the clothes*, *pick up garbage on the ground*, and *open the drawer*. For each task, we report task success under a fixed time budget. Additional dataset/robot details and per-task rubrics are provided in Appendix B.

4.2 Results

Results on LIBERO. Table 2 shows that SimVLA achieves the highest reported average success rate among the compared baselines. Despite using a compact 0.5B backbone, it surpasses large baselines (≥4B) such as OpenVLA-OFT (97.1%) and MemoryVLA (96.7%), demonstrating the strength of a simple, well-tuned architecture. SimVLA achieves near-perfect scores on *Spatial* (99.6%), *Object* (99.8%), and *Goal* (98.6%), ranking first overall. On the challenging *Long* suite, it attains 96.4%, demonstrating strong temporal consistency without explicit memory modules.

Robustness on LIBERO-PRO. Table 3 presents our evaluation on the rigorous LIBERO-PRO benchmark. SimVLA demonstrates strong generalization, particularly under **Semantic** and **Task** perturbations. While OpenVLA and $\pi_{0.5}$ drop to near-zero performance on task-level changes, SimVLA retains nonzero generalization, reaching **10.0%** success on both *Goal* and *Long* suites. SimVLA also ranks first across all suites in Semantic robustness with ∼99% success and performs best on Spatial Position robustness (29.0% in *Spatial*). SimVLA is, however, clearly weaker under Object and Position perturbations on the long-horizon suites (Object-Long 61.0%, Position-Long 3.0%). We do not attribute this gap to a single component, since $\pi_{0.5}$ differs from SimVLA simultaneously in backbone scale, robot-data VLA pretraining, architecture, and training recipe. Disentangling these factors is future work.

Results on WidowX. As detailed in Table 5, SimVLA achieves a 95.8% overall average, matching the heavily pre-trained X-VLA while using only benchmark-specific training from a pretrained VLM initialization. It also secures perfect scores (100%) on two displayed tasks and substantially exceeds large-scale baselines such as MemoryVLA (71.9%) and FPC-VLA (64.6%).

Results on Google Robot. In the Google Robot variant aggregation tasks shown in Table 6, SimVLA achieves 77.0% average success, outperforming strong baselines such as SpatialVLA (67.5%), RT-2-X (65.6%), and ThinkAct (65.1%). SimVLA is also slightly higher than X-VLA (75.7%) on the overall average, although the margin is small. Together, these results suggest that a simple, data-efficient baseline can remain competitive on challenging benchmarks without extensive VLA pretraining.

Results on CALVIN. Table 4 shows that SimVLA achieves the best overall average successful sequence length of 4.47 on CALVIN ABC→D, slightly above compact baselines such as VLA-

Table 7: **Ablations on LIBERO.** Each row corresponds to one ablation setting with the remaining knobs fixed to the default configuration. The default row matches the SimVLA entry in Table 2 and serves as the reference point for the controlled one-knob-at-a-time comparisons. Three-seed std.: Appendix Table 24.

Knob	Value	Spatial	Object	Goal	Long	Avg
SimVLA	Default settings	**99.6**	**99.8**	**98.6**	**96.4**	**98.6**
Data & representation (default: H=10, shuffling on, normalization on)						
Action chunk horizon H	$H = 20$	99.2	89.6	92.4	88.4	92.4
	$H = 30$	95.4	93.8	80.6	79.2	87.3
Data shuffling	off	6.2	0.0	13.6	0.0	5.0
Action normalization	off	22.6	3.2	23.2	0.0	12.3
Optimization dynamics (default: lr 2×10^{-4}, warm-up none, scheduler none, VLM LR multiplier 0.1)						
Learning rate	5×10^{-5}	98.0	97.6	96.2	70.4	90.6
	1×10^{-4}	99.6	98.2	98.4	85.6	95.5
	5×10^{-4}	84.4	91.8	76.0	38.4	72.7
Warm-up steps	1000	97.4	99.6	96.4	93.8	96.8
Scheduler	cosine	99.2	99.4	98.4	93.0	97.5
VLM LR multiplier	1.0	41.2	80.8	46.4	8.2	44.2
Architecture configuration (default: large head (1024,24,16), concat injection, SmolVLM-0.5B, flow matching)						
Action transformer scale	small (768,12,12)	98.8	99.6	98.6	94.8	98.0
Condition injection	conditional AdaLN	99.2	98.0	96.6	70.4	91.1
	cross-attention	98.4	96.6	94.8	76.2	91.5
VLM backbone	Florence-2	99.8	99.2	98.0	93.8	97.7
Action objective	DDPM diffusion	96.6	99.4	96.2	89.8	95.5
	MSE regression	97.2	96.2	96.4	91.2	95.3

Adapter (4.42) and X-VLA (4.43), while also surpassing larger models including OpenVLA-OFT (4.10) and UniVLA (3.80). Despite using only a small backbone, SimVLA reaches 99.4% / 96.4% / 89.8% success on completing 1/2/3 consecutive subtasks, indicating strong long-horizon consistency in the unseen D environment.

Open-world evaluation (BEHAVIOR-1K). As preliminary evidence beyond fixed-task benchmarks, we evaluate SimVLA on the BEHAVIOR-1K Challenge [Li et al., 2024a]. On the scored Task 1 (*Turning on Radio*), SimVLA achieves a task success score of 0.40, outperforming $\pi_{0.5}$ and approaching the 1st-place challenge winner [Larchenko et al., 2025]. Full details are in the Appendix G.

4.3 Ablation Studies

Cross-benchmark attribution hierarchy. We conduct one-knob-at-a-time ablations, varying each factor while holding all other settings fixed. Table 7 reports the main LIBERO audit; Appendix Tables 11–13 extend the same protocol to CALVIN, WidowX, and Google Robot. Across benchmarks, training dynamics (data shuffling, action normalization, VLM LR multiplier) produce the largest swings ($\Delta\approx$54–94%), followed by task configuration ($\Delta\approx$11–43%) and architecture ($\Delta\approx$0.6–15.6%). These deltas are measured relative to our default configuration and do not exhaustively characterize knob interactions, but the consistent ordering is the paper's central empirical finding.

Appendix H repeats the sweeps around alternative base configurations (a small action head, a Florence-2 backbone, and an autoregressive action head) and reports three-seed results for the default SimVLA and for Table 7; the overall ordering is unchanged.

Attribution findings and control implications. The hierarchy implies that VLA comparisons should first match the data pipeline (shuffling and per-dimension normalization), optimizer/backbone adaptation (LR and VLM LR multiplier), and action horizon before attributing gains to architecture. In practice, we recommend prioritizing strong shuffling and normalization, using an LR around 1×10^{-4} with a reduced VLM LR multiplier, matching H to task complexity, and then comparing action objectives or architectural variants against the matched SimVLA baseline.

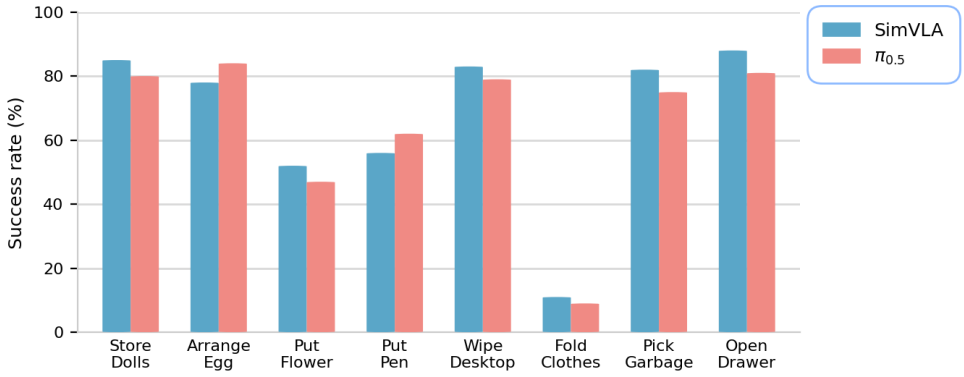


Figure 3: **Real-robot cross-scene zero-shot results on Galaxea R1 Lite.**

Results on Real Robot. On the eight Galaxea R1 Lite tasks, SimVLA is broadly comparable to $\pi_{0.5}$ under the same cross-scene zero-shot protocol (Fig. 3). Most tasks reach around 80% success, while *fold the clothes*, *put the pen into the pen holder*, and *put the flowers in the vase* remain challenging. SimVLA is trained directly from a pretrained VLM, whereas $\pi_{0.5}$ uses its released VLA initialization.

5 Conclusion

We used **SimVLA**—a minimal VLM encoder plus lightweight continuous-action head—as a controlled testbed for attributing VLA performance gains. Across LIBERO, CALVIN, WidowX, and Google Robot, the same hierarchy appears: training dynamics have the largest observed effect ($\Delta\approx$54–94%), task configuration ranks second ($\Delta\approx$11–43%), and architecture has the smallest measured effect ($\Delta\approx$0.6–15.6%). SimVLA also achieves leading or competitive performance across the studied benchmarks, showing that careful benchmark-specific training from a pretrained VLM can rival larger VLA-pretrained systems. We therefore propose SimVLA as a matched testbed: future modules should be added while holding training dynamics and task configuration fixed, and gains should be reported as deltas against this baseline. Whether the hierarchy transfers to substantially different VLA families remains open.

Limitations

SimVLA is intended as a controlled attribution testbed rather than a universal VLA recipe. Its conclusions are therefore strongest for manipulation-oriented VLM-encoder plus lightweight-head policies, where matched training dynamics and task settings can be cleanly compared; Extending the same analysis to broader embodiments, task families, and deployment metrics is an important direction for future work.

Broader Impact

By clarifying which factors drive VLA performance, SimVLA may make robotic policy research more reproducible and accessible by reducing unnecessary architectural and pretraining costs. At the same time, stronger manipulation policies could accelerate deployment in settings where safety validation, human oversight, and labor impacts require careful assessment. We view SimVLA as an evaluation and ablation tool, not a recommendation for unsupervised deployment in safety-critical environments.

Acknowledgements and Disclosures

This work received support from the “111 Center” (No. B26023) and NSFC Grant 62606225.

References

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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

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, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

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

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Yandu Chen, Kefan Gu, Yuqing Wen, Yucheng Zhao, Tiancai Wang, and Liqiang Nie. Intentionvla: Generalizable and efficient embodied intention reasoning for human-robot interaction. *arXiv preprint arXiv:2510.07778*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu,

Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

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

Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, and Lin Shao. Vla-os: Structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv preprint arXiv:2506.17561*, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

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

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

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. Openvla: An open-source Vision-Language-Action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Hamlet: Switch your vision-language-action model into a history-aware policy. *arXiv preprint arXiv:2510.00695*, 2025.

Ilia Larchenko, Gleb Zarin, and Akash Karnatak. Task adaptation of vision-language-action model: 1st place solution for the 2025 behavior challenge, 2025. URL https://arxiv.org/abs/2512.06951.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation, 2024a. URL https://arxiv.org/abs/2403.09227.

Xinghang Li, Peiyan Li, Long Qian, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Xinlong Wang, Di Guo, Tao Kong, Hanbo Zhang, and Huaping Liu. What matters in building vision-language-action models for generalist robots, 2024b. URL https://arxiv.org/abs/2412.14058.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation, 2024c. URL https://arxiv.org/abs/2405.05941.

Wenqi Liang, Gan Sun, Yao He, Jiahua Dong, Suyan Dai, Ivan Laptev, Salman Khan, and Yang Cong. Pixelvla: Advancing pixel-level understanding in vision-language-action model. *arXiv preprint arXiv:2511.01571*, 2025a.

Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025b.

Minghui Lin, Pengxiang Ding, Shu Wang, Zifeng Zhuang, Yang Liu, Xinyang Tong, Wenxuan Song, Shangke Lyu, Siteng Huang, and Donglin Wang. Hif-vla: Hindsight, insight and foresight through motion representation for vision-language-action models. *arXiv preprint arXiv:2512.09928*, 2025.

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

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

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. URL https://arxiv.org/abs/2504.05299.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

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

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

Qwen Team. Qwen3-vl technical report, 2025. TODO: confirm final citation source before submission.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

Ranjan Sapkota, Yang Cao, Konstantinos I. Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges, 2025. URL https://arxiv.org/abs/2505.04769.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Zhijun Li, Donglin Wang, Jun Ma, Lujia Wang, and Haoang Li. Pd-vla: Accelerating vision-language-action model integrated with action chunking via parallel decoding, 2025a. URL https://arxiv.org/abs/2503.02310.

Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333*, 2025b.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

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. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. *arXiv preprint arXiv:2507.01016*, 2025a.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025b.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025c.

Xiao-Ming Wu, Bin Fan, Kang Liao, Jian-Jian Jiang, Runze Yang, Yihang Luo, Zhonghua Wu, Wei-Shi Zheng, and Chen Change Loy. Vlanext: Recipes for building strong vla models, 2026. URL https://arxiv.org/abs/2602.18532.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

Chao Xu, Suyu Zhang, Yang Liu, Baigui Sun, Weihong Chen, Bo Xu, Qi Liu, Juncheng Wang, Shujun Wang, Shan Luo, Jan Peters, Athanasios V. Vasilakos, Stefanos Zafeiriou, and Jiankang Deng. An anatomy of vision-language-action models: From modules to milestones and challenges, 2025. URL https://arxiv.org/abs/2512.11362.

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Piaopiao Jin, Guokang Sun, Shaoqing Xu, Yangwei You, and Jingtai Liu. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *arXiv preprint arXiv:2509.04018*, 2025.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. 4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration. *arXiv preprint arXiv:2506.22242*, 2025a.

Kun Zhang, Peng Yun, Jun Cen, Junhao Cai, Didi Zhu, Hangjie Yuan, Chao Zhao, Tao Feng, Michael Yu Wang, Qifeng Chen, Jia Pan, Wei Zhang, Bo Yang, and Hua Chen. Generative artificial intelligence in robotic manipulation: A survey, 2025b. URL https://arxiv.org/abs/2503.03464.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025c.

Zhengshen Zhang, Hao Li, Yalun Dai, Zhengbang Zhu, Lei Zhou, Chenchen Liu, Dong Wang, Francis E. H. Tay, Sijin Chen, Ziwei Liu, Yuxiao Liu, Xinghang Li, and Pan Zhou. From spatial to actions: Grounding vision-language-action model in spatial foundation priors. *arXiv preprint arXiv:2510.17439*, 2025d.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *CVPR 2025*, 2025. arXiv:2503.22020.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. *arXiv preprint arXiv:2501.10105*, 2025a.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025b.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daum'e, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *ArXiv*, abs/2412.10345, 2024.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. Flowvla: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

Appendix

A Baselines

Benchmark comparison baselines. We compare SimVLA against a spectrum of 34 representative policies in the benchmark tables, ranging from standard baselines to recent complex architectures: RT-1-X / RT-2-X [Collaboration et al., 2025], OpenVLA [Kim et al., 2024], Octo-Small / Octo-Base [Team et al., 2024], TraceVLA [Zheng et al., 2024], SpatialVLA [Qu et al., 2025], UnifiedVLA [Wang et al., 2025c], UniVLA [Bu et al., 2025], X-VLA [Zheng et al., 2025b], VLA-Adapter [Wang et al., 2025b], VLA-OS [Gao et al., 2025], UniAct [Zheng et al., 2025a], NORA [Hung et al., 2025], MemoryVLA [Shi et al., 2025], ThinkAct [Huang et al., 2025], CoT-VLA [Zhao et al., 2025], WorldVLA [Cen et al., 2025], SmolVLA [Shukor et al., 2025], MolmoAct [Lee et al., 2025], π_0 [Black et al., 2024], π_0-FAST [Pertsch et al., 2025], $\pi_{0.5}$ [Black et al., 2025], DD-VLA [Liang et al., 2025b], OpenVLA-OFT [Kim et al., 2025], RoboVLM [Li et al., 2024b], GR00T-N1 [NVIDIA et al., 2025], FlowVLA [Zhong et al., 2025], PD-VLA [Song et al., 2025a], FPC-VLA [Yang et al., 2025], 4D-VLA [Zhang et al., 2025a], GraspVLA [Deng et al., 2025], FLOWER [Reuss et al., 2025], VLANeXt [Wu et al., 2026], HiF-VLA [Lin et al., 2025], Seer [Tian et al., 2024].

Related-work categorization. Table 8 maps the baselines discussed in Section 2 to their corresponding design axis. We organize existing VLA methods along three axes: (1) *Visual and Temporal Augmentation*, which adds auxiliary visual cues such as flow, traces, or chain-of-thought reasoning; (2) *Geometric and 3D Priors*, which injects spatial structure via depth, point clouds, or 3D representations; and (3) *Action Representations and Architectures*, which focuses on the action decoding mechanism itself (e.g., diffusion, flow matching, autoregressive tokenization, or adapter-based heads). The majority of recent baselines fall into Category 3, reflecting the community's focus on action-head design. SimVLA achieves leading performance without the auxiliary modules from Categories 1 and 2, suggesting that a well-tuned training recipe applied to a simple architecture can match or exceed methods that rely on additional visual or spatial priors.

Table 8: **Categorization Summary.** The baselines are mapped to their corresponding design axis as defined in Section 2 (Related Work).

Design Axis	Models
(1) Visual and Temporal Augmentation	FlowVLA [Zhong et al., 2025], CoT-VLA [Zhao et al., 2025], TraceVLA [Zheng et al., 2024], 4D-VLA [Zhang et al., 2025a], ThinkAct [Huang et al., 2025], MemoryVLA [Shi et al., 2025], FPC-VLA [Yang et al., 2025]
(2) Geometric and 3D Priors	MolmoAct [Lee et al., 2025], SpatialVLA [Qu et al., 2025], GraspVLA [Deng et al., 2025], 4D-VLA [Zhang et al., 2025a]
(3) Action Representations and Architectures	OpenVLA-OFT [Kim et al., 2025], DD-VLA [Liang et al., 2025b], PD-VLA [Song et al., 2025a], UniVLA [Bu et al., 2025], UnifiedVLA [Wang et al., 2025c], UniAct [Zheng et al., 2025a], π_0 [Black et al., 2024], π_0-FAST [Pertsch et al., 2025], $\pi_{0.5}$ [Black et al., 2025], NORA [Hung et al., 2025], SmolVLA [Shukor et al., 2025], GR00T-N1 [NVIDIA et al., 2025], FLOWER [Reuss et al., 2025], X-VLA [Zheng et al., 2025b], VLA-Adapter [Wang et al., 2025b], VLA-OS [Gao et al., 2025], WorldVLA [Cen et al., 2025]

B Galaxea Open-World Dataset

We train on the Galaxea Open-World Dataset [Jiang et al., 2025], a large-scale real-world mobile-manipulation dataset with $\sim$500 hours of demonstrations (about 100K trajectories). The dataset spans 150 task categories across 50 real-world scenes and covers more than 1,600 objects and 58 skills. Importantly for our setting, data are collected under a single consistent embodiment, so that perception streams, action/state signals, and language annotations are naturally aligned for end-to-end VLA training.

Platform. All demonstrations are recorded on the Galaxea R1 Lite platform, which is also the real robot we use for our cross-scene zero-shot evaluation in Sec. 4. R1 Lite is a mobile bimanual robot with a 23-DoF embodiment (two 6-DoF arms, a 3-DoF torso for workspace extension, a 6-DoF vector-drive omnidirectional base up to 1.5 m/s, and two 1-DoF grippers). The platform is equipped with a head stereo RGB camera for scene-level context and dual Intel RealSense D405 RGB-D wrist cameras for close-range manipulation.

C Real-Robot Evaluation Details

For each real-robot task, we use a simple rubric that measures whether the policy can complete the task within the fixed time budget used in Sec. 4. We report binary success/failure over 50 trials.

Store the dolls. The robot must pick up **three** plush dolls placed on a tabletop and put them into a designated storage container. Success is defined as all three dolls being fully inside the container at the end of the episode.

Arrange eggs. The robot must pick up **one** egg and place it into a designated slot of an egg carton. Success is defined as the egg being seated in the target slot without being dropped.

Put the flowers in the vase. The robot must grasp a flower and insert it into the opening of a vase. Success is defined as the flower being inside the vase and remaining stably placed at the end of the episode.

Put the pen into the pen holder. The robot must pick up **two** pens and insert them into a pen holder. Success is defined as both pens being inside the holder (stably contained) at the end of the episode.

Wipe the desktop. The robot must wipe a visible stain/dirty region on a desktop using a **cloth rag**. Success is defined as the target region being visibly cleaned (stain disappears or is substantially reduced) within the time budget.

Fold the clothes. The robot must fold a shirt starting from a flat or mildly wrinkled state on the table. Success is defined as completing the target fold (folding in sleeves and making a main fold) with the garment remaining on the working surface.

Pick up garbage on the ground. The robot must pick up **two** plastic bottles from the ground and dispose them into a trash bin. Success is defined as both bottles ending inside the bin.

Open the drawer. The robot must grasp the drawer handle and pull to open the drawer to a specified extent. Success is defined as the drawer being opened beyond a threshold without damaging the mechanism.

D Training Hyperparameters

To facilitate reproducibility, we summarize the key hyperparameters used in our simulation training (LIBERO, CALVIN, WidowX, Google Robot, and BEHAVIOR-1K) and real-robot training (Galaxea-500h). All simulation runs are executed on **4**×**H100** GPUs, except BEHAVIOR-1K which uses **8**×**H100**, and are initialized from a pretrained VLM backbone without additional large-scale robot-data VLA pretraining. For comprehensive implementation details, please refer to our codebase.

Simulation. Table 9 reports the configuration.

Real-robot training (Galaxea-500h). Table 10 summarizes the training setup on the Galaxea Open-World Dataset. For a fair comparison, both our method and $\pi_{0.5}$ are trained using the same Galaxea data and the same hyperparameters, with the same compute budget: **64**×**H100** GPUs for **150K** training steps.

Table 9: **Training Hyperparameters and Inference Settings for Simulation Datasets.**

Configuration	Libero	WidowX	Google Robot	Calvin	BEHAVIOR-1K
Batch size (per GPU)	64	80	80	48	32
Global batch size	64×4 = 256	80×4 = 320	80×4 = 320	48×4 = 192	32×8 = 256
Action head	Large (1024,24,16)	Large (1024,24,16)	Large (1024,24,16)	Large (1024,24,16)	Large (1024,24,16)
Action chunk horizon H	10	30	30	30	30
Action objective	conditional flow matching	conditional flow matching	conditional flow matching	conditional flow matching	conditional flow matching
Noise distribution	standard Gaussian	standard Gaussian	standard Gaussian	standard Gaussian	standard Gaussian
Time sampling	Beta$(1.5, 1)$ on $[0.001, 1]$	Beta$(1.5, 1)$ on $[0.001, 1]$	Beta$(1.5, 1)$ on $[0.001, 1]$	Beta$(1.5, 1)$ on $[0.001, 1]$	Beta$(1.5, 1)$ on $[0.001, 1]$
Training target	velocity $(\epsilon - x)$	velocity $(\epsilon - x)$	velocity $(\epsilon - x)$	velocity $(\epsilon - x)$	velocity $(\epsilon - x)$
Inference solver	Euler integration	Euler integration	Euler integration	Euler integration	Euler integration
Inference steps	10	10	10	10	10
Image resize	128×128	224×224	224×224	224×224	224×224
Action normalization	on	on	on	on	on
Data shuffling	on	on	on	on	on
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Betas	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Weight decay	0.0	0.0	0.0	0.0	0.0
Learning rate	2e-4	1e-4	1e-4	1e-4	1e-4
VLM LR multiplier	0.1	0.05	0.01	0.1	0.01
Warm-up steps	0	2000	2000	2000	2000
Scheduler	none	none	none	cosine	none
Training steps	150K	100K	150K	150K	300K
Precision	bfloat16	bfloat16	bfloat16	bfloat16	bfloat16

Table 10: **Hyperparameters for Galaxea-500h Training.**

Configuration	Value
Batch size (per GPU)	32
Global batch size	32×64=2048
Action head	Large (1024,24,16)
Action chunk horizon H	30
Image resize	224×224
Action normalization	on
Data shuffling	on
Optimizer	AdamW
Betas	(0.9, 0.95)
Weight decay	0.0
Learning rate	1e-4
VLM LR multiplier	0.1
Warm-up steps	1000
Scheduler	none
Training steps	150K
Precision	bfloat16

E Per-Benchmark Ablation Breakdowns

The main text (Table 7) presents detailed ablations on LIBERO. Tables 11–13 below extend the same controlled protocol to CALVIN, WidowX, and Google Robot. In each table, one knob is varied while all others are held at the default configuration. The consistent factor ordering across all three additional benchmarks—training dynamics causing the largest performance swings, followed by task configuration, then architecture—corroborates the attribution hierarchy established in Section 4.3.

CALVIN ABC→D. Table 11 reports ablations under the standard ABC→D protocol, where the policy is trained on scenes A, B, C and evaluated on 1,000 five-step language-conditioned chains in the held-out scene D. CALVIN is particularly sensitive to data pipeline choices: disabling shuffling drops the score from 4.47 to 0.17, and setting the VLM LR multiplier to 1.0 yields a similarly sharp drop to 0.16. The action horizon also plays a notable role (H=10 drops the score to 3.84), confirming that longer horizons are beneficial for multi-step sequential tasks. For this horizon ablation we changed only the action chunk horizon, from the default H=30 to H=10.

WidowX (SimplerEnv-Bridge). Table 12 reports ablations on the four WidowX tabletop tasks evaluated via SimplerEnv. The same hierarchy holds: disabling shuffling drops the average from 95.8% to 37.5%, and setting the VLM LR multiplier to 1.0 causes a sharp drop to 11.5%. Architecture changes are again secondary—switching to a small action head reduces performance to 80.2%, a much smaller swing than the training-dynamics knobs.

Table 11: **Ablations on CALVIN.** Each row corresponds to one ablation setting with the remaining knobs fixed to the default configuration.

Knob	Value	SR1	SR2	SR3	SR4	SR5	Score
SimVLA	Default settings	**99.4**	**96.4**	**89.8**	**83.7**	**77.2**	**4.47**
Data & representation (default: H=30, shuffling on, normalization on)							
Action chunk horizon H	$H = 10$	96.9	86.7	76.6	66.7	56.8	3.84
Data shuffling	off	14.7	2.2	0.4	0.1	0.0	0.17
Action normalization	off	95.4	84.7	73.7	63.5	53.7	3.73
Optimization dynamics (default: lr 1×10^{-4}, warm-up 2000, scheduler cosine, VLM LR multiplier 0.1)							
Learning rate	5×10^{-5}	98.4	92.4	83.5	71.8	63.3	4.10
Warm-up steps	1000	97.7	91.3	82.2	71.0	62.2	4.05
Scheduler	no cosine	97.8	90.7	80.3	68.7	58.6	3.96
VLM LR multiplier	1.0	14.8	1.4	0.0	0.0	0.0	0.16
Architecture configuration (default: large head (1024,24,16), concat injection)							
Action transformer scale	small (768,12,12)	95.7	90.3	78.1	68.1	59.2	3.91

Table 12: **Ablations on WidowX.** Each row corresponds to one ablation setting with the remaining knobs fixed to the default configuration.

Knob	Value	spoon	carrot	stack	eggplant	Avg
SimVLA	Default settings	**100.0**	**100.0**	**91.7**	**91.7**	**95.8**
Data & representation (default: H=30, shuffling on, normalization on)						
Action chunk horizon H	$H = 10$	95.8	83.3	70.8	21.7	67.9
Data shuffling	off	54.2	50.0	29.2	16.7	37.5
Action normalization	off	79.2	95.8	79.2	34.8	72.3
Optimization dynamics (default: lr 1×10^{-4}, warm-up 2000, scheduler none, VLM LR multiplier 0.05)						
Learning rate	5×10^{-5}	95.8	91.7	91.7	75.0	88.6
Warm-up steps	1000	95.8	95.8	91.7	75.0	89.6
Scheduler	cosine	95.8	95.8	91.7	79.2	90.6
VLM LR multiplier	1.0	29.2	16.7	0.0	0.0	11.5
Architecture configuration (default: large head (1024,24,16), concat injection)						
Action transformer scale	small (768,12,12)	70.8	95.8	100.0	54.2	80.2

Google Robot (SimplerEnv-Fractal). Table 13 reports ablations on the Google Robot variant-aggregation tasks (Pick, Move Near, Open/Close Drawer). The action horizon effect is especially pronounced here: reducing H from 30 to 10 drops the average from 77.0% to 34.2%, the largest horizon-induced swing across all four benchmarks. Training dynamics remain the largest measured factor—disabling shuffling (-24.0 pp) and setting VLM LR multiplier to 1.0 (-21.0 pp) cause large degradations—while the small action head is within 1.6 pp of the default.

F Qualitative Rollouts

Figures 4–6 show representative rollout sequences from three simulation benchmarks, illustrating the diversity of manipulation skills that SimVLA executes successfully under its default configuration. Each figure displays keyframes from a successful episode, demonstrating that SimVLA produces smooth, temporally consistent action trajectories across diverse manipulation scenarios.

Table 13: **Ablations on Google Robot (Variant Aggregation).** Each row corresponds to one ablation setting with the remaining knobs fixed to the default configuration.

Knob	Value	Pick	Move	Open	Avg
SimVLA	Default settings	**82.3**	**81.0**	**67.7**	**77.0**
Data & representation (default: H=30, shuffling on, normalization on)					
Action chunk horizon H	$H = 10$	29.0	32.2	41.3	34.2
Data shuffling	off	61.2	53.8	43.9	53.0
Action normalization	off	72.5	67.8	64.8	68.4
Optimization dynamics (default: lr 1×10^{-4}, warm-up 2000, scheduler none, VLM LR multiplier 0.01)					
Learning rate	5×10^{-5}	81.3	73.3	67.7	74.1
Warm-up steps	1000	75.0	78.2	66.1	73.1
Scheduler	cosine	79.8	77.0	67.9	74.9
VLM LR multiplier	1.0	62.8	50.3	54.8	56.0
Architecture configuration (default: large head (1024,24,16), concat injection)					
Action transformer scale	small (768,12,12)	81.9	78.3	66.1	75.4

Table 14: **BEHAVIOR-1K Challenge** task success scores under the challenge protocol (open-world, randomized poses and objects).

Task	**SimVLA**	$\pi_{0.5}$	1st-place winner [Larchenko et al., 2025]
Task 1: Turning on Radio	**0.40**	0.00	0.60
Task 2: Picking Up Trash	**0.50**	0.00	1.00

G BEHAVIOR-1K Challenge Results

To assess whether SimVLA's competence extends beyond fixed-task simulation benchmarks, we conduct a preliminary evaluation on the BEHAVIOR-1K Challenge [Li et al., 2024a]. BEHAVIOR-1K tests open-world robustness: the robot operates in a full household environment with randomized object layouts, initial poses, and scene configurations. Table 14 reports results on Task 1 (*Turning on Radio*), where the agent must navigate to a radio, interact with it, and successfully turn it on under randomized conditions. SimVLA achieves a task success score of 0.40, outperforming $\pi_{0.5}$ (3B backbone, pretrained on 200k+ robot episodes), which scores 0.00, and approaching the 1st-place challenge winner (0.60) [Larchenko et al., 2025]. This result is notable because SimVLA uses only a small backbone and no additional large-scale robot-data VLA pretraining before benchmark-specific training, yet remains competitive in an open-world setting designed to stress-test generalization beyond the controlled environments of LIBERO or CALVIN. SimVLA also succeeds on Task 2 (*Picking Up Trash*)—a 3–5 minute long-horizon episode requiring the robot to navigate from a random starting pose, search for scattered trash, grasp each piece, and place it in a bin. Training and inference settings are reported in Table 9.

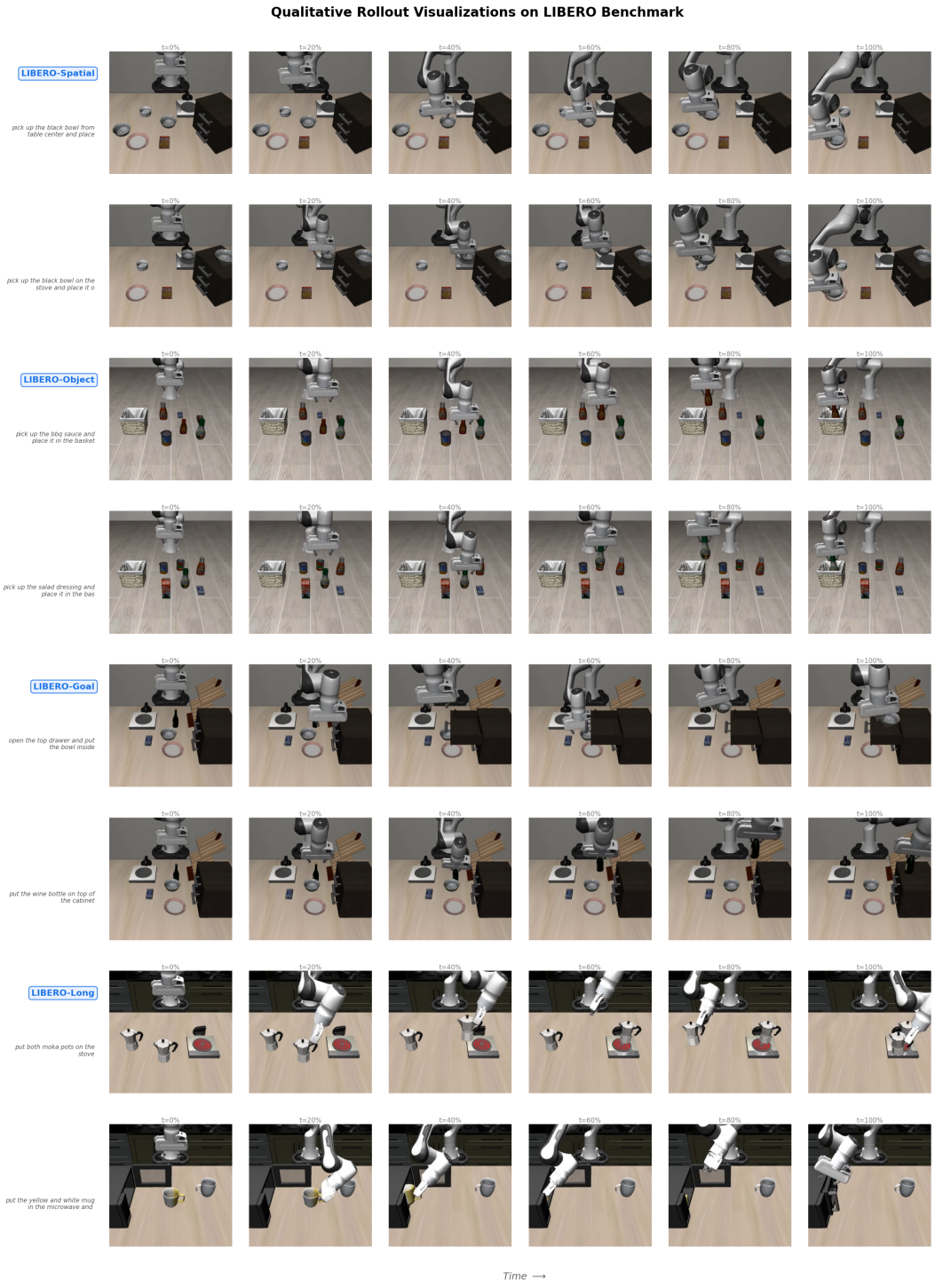


Figure 4: **Qualitative Rollout on LIBERO.** Keyframes from a representative long-horizon kitchen manipulation task. The policy must execute a sequence of precise pick-and-place actions in the correct order. SimVLA completes the full multi-step sequence with smooth transitions between subtasks, reflecting strong temporal consistency under the $H=10$ action chunking horizon used for LIBERO.

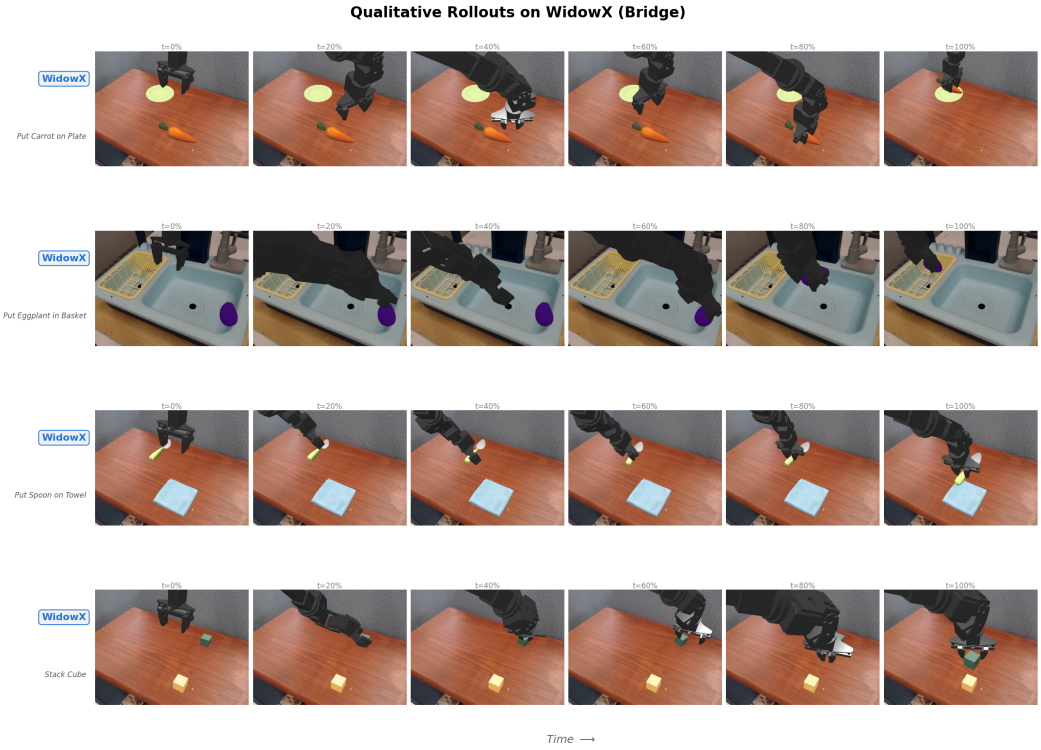


Figure 5: **Qualitative Rollout on WidowX.** Keyframes from a tabletop pick-and-place task in the SimplerEnv-Bridge environment. The WidowX robot must locate, grasp, and transport an object to a target location from a single third-person RGB view. SimVLA achieves 95.8% average success on this benchmark, matching X-VLA without additional large-scale robot-data VLA pretraining before benchmark-specific training.

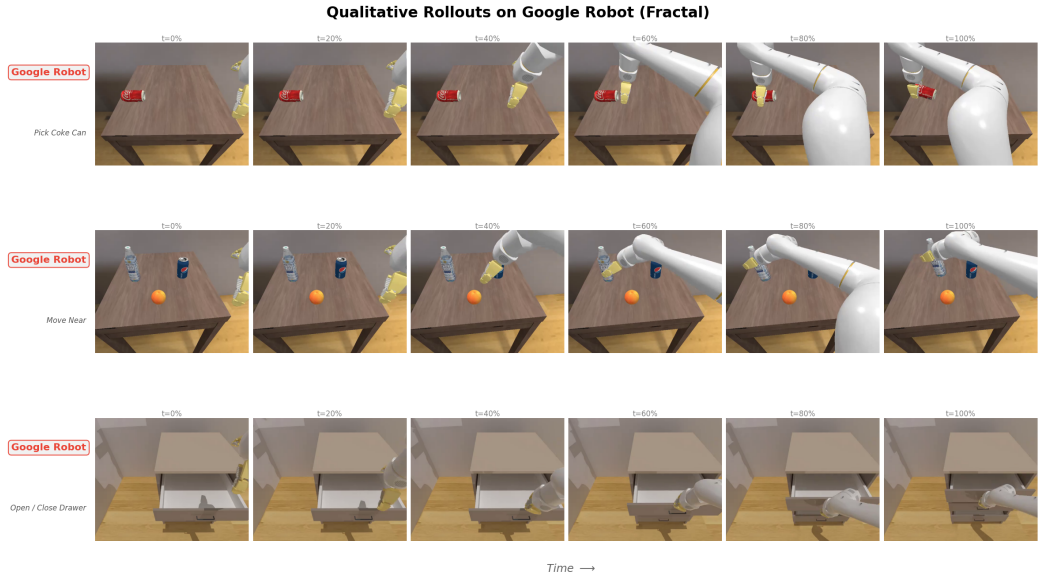


Figure 6: **Qualitative Rollout on Google Robot.** Keyframes from a drawer manipulation task in the SimplerEnv-Fractal environment. The policy controls a mobile manipulator to approach and operate a drawer handle. SimVLA achieves 77.0% variant-aggregation average across Pick, Move Near, and Open/Close Drawer tasks on this benchmark.

Table 15: **LIBERO one-knob-at-a-time sweep at the small-head base** ({768,12,12}). Success rate (%).

Knob	Value	Spatial	Object	Goal	Long	Avg
SimVLA	Default	98.8	99.6	98.6	94.8	98.0
Horizon	$H=20$	96.4	96.6	93.4	79.8	91.6
	$H=30$	97.4	99.2	78.6	75.4	87.7
Shuffling	off	10.8	0.0	14.2	0.0	6.3
Norm	off	13.4	8.6	23.8	0.0	11.4
LR	5×10^{-5}	99.4	99.4	95.4	70.4	91.2
	1×10^{-4}	98.8	99.6	98.6	94.2	97.8
	5×10^{-4}	89.0	99.0	59.0	64.0	77.8
Warm-up	1000	99.4	99.0	98.0	90.8	96.8
Scheduler	cosine	99.0	98.2	97.2	91.6	96.5
VLM LR mult	1.0	64.4	78.2	73.8	13.6	57.5
Injection	AdaLN	98.2	99.2	96.7	80.0	93.5
	cross-attn	98.2	98.6	92.4	82.4	92.9
Backbone	Florence-2	99.4	99.0	98.0	90.8	96.8

Table 16: **WidowX one-knob-at-a-time sweep at the small-head base.** Success rate (%).

Knob	Value	Spoon	Carrot	Stack	Eggplant	Avg
SimVLA	Default	70.8	95.8	100.0	54.2	80.2
Horizon	$H=10$	70.8	79.2	83.3	16.7	62.5
Shuffling	off	41.7	50.0	33.3	0.0	31.3
Norm	off	66.7	95.8	83.3	0.0	61.5
LR	5×10^{-5}	95.8	95.8	100.0	58.3	87.5
Warm-up	1000	91.7	100.0	87.5	20.8	75.0
Scheduler	cosine	83.3	95.8	95.8	41.7	79.2
VLM LR mult	1.0	16.7	16.7	4.2	0.0	9.4

H Robustness of the Attribution Hierarchy

This appendix provides additional experiments that support the robustness of the attribution hierarchy of Section 4.3 (training dynamics > task configuration > architecture). Because the main-text ablations vary one knob at a time around a single default, we examine whether the ordering (i) persists when the base configuration changes (Section H.1), (ii) extends to a different action-generation family (Section H.2), (iii) is large relative to seed-to-seed variation (Section H.3), and (iv) is stable when a third backbone family is added (Section H.4). Unless stated otherwise, all runs follow the protocol of Section 3.3: one knob is changed at a time and every other setting stays at the reference value of the respective base configuration. We call the reference configuration of the main text the *default base*.

Table 17: **CALVIN ABC→D one-knob-at-a-time sweep at the small-head base.** "Scheduler: no cosine" removes the default cosine decay.

Knob	Value	SR1	SR2	SR3	SR4	SR5	Avg Length
SimVLA	Default	95.7	90.3	78.1	68.1	59.2	3.91
Horizon	$H{=}10$	88.5	79.0	69.8	60.9	51.8	3.50
Shuffling	off	18.6	2.1	0.3	0.0	0.0	0.21
Norm	off	89.0	79.0	68.7	59.2	50.1	3.46
LR	5×10^{-5}	99.4	95.8	90.2	85.0	76.6	4.47
Warm-up	1000	99.0	94.4	86.6	79.2	68.0	4.27
Scheduler	no cosine	98.6	95.8	91.0	85.2	75.8	4.46
VLM LR mult	1.0	16.9	1.1	0.0	0.0	0.0	0.18

Table 18: **LIBERO one-knob-at-a-time sweep at the Florence-2 (0.9B) base** with the large action head. Success rate (%).

Knob	Value	Spatial	Object	Goal	Long	Avg
SimVLA	Default	99.8	99.2	98.0	93.8	97.7
Horizon	$H{=}20$	97.4	95.4	90.6	87.2	92.6
	$H{=}30$	97.2	91.6	88.2	68.8	86.5
Shuffling	off	7.8	0.0	26.4	0.0	8.6
Norm	off	20.8	0.0	26.4	0.0	11.8
LR	5×10^{-5}	95.1	99.2	93.4	82.6	92.6
	1×10^{-4}	99.0	98.2	97.2	91.6	96.5
	5×10^{-4}	95.8	84.6	84.4	49.2	78.5
Warm-up	1000	99.4	98.0	97.0	90.2	96.1
Scheduler	cosine	99.8	99.2	98.0	93.0	97.5
VLM LR mult	1.0	76.4	99.2	13.8	13.4	50.7
Injection	AdaLN	98.6	96.0	92.2	74.4	90.3
	cross-attn	98.6	96.2	96.0	72.4	90.8
Action head	Small	98.2	98.4	98.4	94.6	97.4

H.1 Sweeps around alternative base configurations

The main-text ablations vary one knob around a single default. To test whether the ordering is a property of that default point, we repeat the sweep around further bases:

- **Small-head base on three benchmarks:** small action head $\{768, 12, 12\}$ on LIBERO (Table 15), WidowX (Table 16) and CALVIN (Table 17).
- **Florence-2 base on LIBERO:** the Florence-2 backbone (Table 18).

Table 19: **Autoregressive-head experiment on LIBERO.** The continuous flow-matching head is replaced by a discrete autoregressive action head (256 bins) on the same SmolVLM-0.5B encoder; the AR default uses cosine scheduling. Success rate (%).

Knob	Value	Spatial	Object	Goal	Long	Avg
SimVLA (AR)	Default	94.8	99.4	92.6	94.8	95.4
Horizon	$H{=}20$	98.2	95.6	93.8	82.0	92.4
	$H{=}30$	96.0	97.0	81.6	87.0	90.4
Shuffling	off	0.0	0.0	0.0	0.0	0.0
LR	5×10^{-5}	97.4	96.2	91.6	81.8	91.8
	1×10^{-4}	97.2	95.2	91.6	92.2	94.0
	5×10^{-4}	0.0	0.0	0.0	0.0	0.0
Warm-up	1000	97.8	99.8	90.0	86.6	93.5
Scheduler	no cosine	94.4	93.0	65.8	63.8	79.2
VLM LR multiplier	1.0	39.8	60.4	47.2	0.0	36.9
Head scale	small	94.6	98.2	88.0	89.0	92.5

Table 20: **LIBERO, three training seeds.** X-VLA is reproduced with the same three-seed procedure.

Method	Spatial	Object	Goal	Long	Average
SimVLA	**99.4 ± 0.5**	**99.6 ± 0.4**	**98.0 ± 0.6**	**95.4 ± 0.8**	**98.1 ± 0.5**
XVLA	98.2 ± 0.8	98.5 ± 0.5	97.2 ± 0.4	96.3 ± 0.5	97.5 ± 0.4

Table 21: **WidowX, three training seeds.**

Method	Carrot on Plate	Eggplant in Basket	Spoon on Towel	Stack Cube	Average
SimVLA	**97.2 ± 2.4**	**86.1 ± 9.6**	**98.6 ± 2.4**	**93.1 ± 2.4**	**93.8 ± 2.1**

Table 22: **Google Robot, three training seeds.**

Method	Coke Can	Move Near	Open/Close	VA Average
SimVLA	**83.5 ± 1.6**	**77.1 ± 3.4**	**68.2 ± 0.5**	**76.3 ± 0.8**

Table 23: **CALVIN ABC→D, three training seeds.**

Method	Average Length	SR@1 (%)	SR@2 (%)	SR@3 (%)	SR@4 (%)	SR@5 (%)
SimVLA	**4.4 ± 0.1**	**99.3 ± 0.5**	**96.1 ± 0.4**	**89.1 ± 0.7**	**82.9 ± 0.8**	**73.7 ± 3.1**

At every base the largest training-dynamics drop exceeds the largest task-configuration drop, which in turn exceeds the largest architecture drop wherever architecture knobs were swept. The shuffling and horizon deltas are nearly identical across the three LIBERO base configurations (SmolVLM + small head, Florence-2 + large head, plus the SmolVLM + large head default), while the VLM LR multiplier delta varies more (54.4, 40.5, 47.0) but stays far above every task-configuration and architecture delta. The training-dynamics maximum stays at roughly 89–94 points and the task-configuration maximum at roughly 10–11 points across the three LIBERO bases, so the two tiers remain separated by about an order of magnitude at each base. Extending to CALVIN and WidowX at the small head base, the training-dynamics vs. task-configuration ratio of tier-max deltas is 9.1 and 4.0 respectively. Neither the tier ordering flips nor the tier-max ranges overlap on any of the configurations tested.

Table 24: **Table 7 with three training seeds (LIBERO).** Mean ± sample standard deviation over seeds 0, 1, 2. The default row is the three-seed mean of the SimVLA default.

Knob	Value	Spatial	Object	Goal	Long	Avg
SimVLA	Default	99.4 ± 0.5	99.6 ± 0.4	98.0 ± 0.6	95.4 ± 0.8	98.1 ± 0.5
Horizon	H=20	99.4 ± 0.7	88.0 ± 1.3	92.0 ± 1.4	87.8 ± 2.1	91.8 ± 1.0
	H=30	95.8 ± 1.8	92.6 ± 0.9	81.4 ± 2.6	77.8 ± 4.2	86.9 ± 1.6
Shuffling	off	5.4 ± 5.1	0.0 ± 0.0	12.2 ± 7.3	0.0 ± 0.0	4.4 ± 0.0
Norm	off	21.4 ± 6.8	2.6 ± 2.9	24.0 ± 5.4	0.0 ± 0.0	12.0 ± 3.2
LR	5×10^{-5}	97.6 ± 0.6	98.2 ± 1.1	95.4 ± 0.8	69.6 ± 3.7	90.2 ± 1.2
	1×10^{-4}	99.2 ± 0.4	97.6 ± 0.7	98.0 ± 0.5	84.4 ± 2.6	94.8 ± 1.0
	5×10^{-4}	83.8 ± 3.5	92.4 ± 1.6	75.2 ± 4.8	37.8 ± 7.1	72.3 ± 3.4
Warm-up	1000	97.0 ± 1.2	99.4 ± 0.3	96.8 ± 0.9	92.8 ± 1.5	96.5 ± 0.7
Scheduler	cosine	99.0 ± 0.4	99.0 ± 0.6	98.0 ± 0.5	92.0 ± 1.3	97.0 ± 0.6
VLM LR mult	1.0	40.4 ± 8.3	81.2 ± 3.2	45.8 ± 6.9	7.8 ± 3.6	43.8 ± 4.1
Head scale	small	98.4 ± 0.7	99.2 ± 0.3	98.0 ± 0.4	93.6 ± 1.6	97.3 ± 0.6
Injection	AdaLN	99.0 ± 0.5	97.6 ± 0.9	96.2 ± 0.7	69.2 ± 0.9	90.5 ± 0.8
	cross-attn	98.6 ± 0.6	96.2 ± 1.1	95.2 ± 1.0	75.2 ± 1.3	91.3 ± 0.9
Backbone	Florence-2	99.4 ± 0.3	99.0 ± 0.5	97.6 ± 0.4	92.0 ± 1.1	97.0 ± 0.5
Objective	DDPM	96.2 ± 1.4	99.0 ± 0.4	96.4 ± 0.9	89.2 ± 1.7	95.2 ± 0.8
	MSE	96.6 ± 0.8	95.8 ± 1.5	96.0 ± 0.6	89.6 ± 2.0	94.5 ± 1.1

Table 25: **Backbone families on LIBERO (success rate, %).** Qwen3-VL-2B is a third backbone family; Δ is the drop from the SmolVLM default.

Backbone	Params	Spatial	Object	Goal	Long	Avg	Δ vs default
SmolVLM (default)	0.5B	99.6	99.8	98.6	96.4	**98.6**	—
Florence-2	0.9B	99.8	99.2	98.0	93.8	**97.7**	0.9
Qwen3-VL-2B	2B	99.0	98.2	96.6	94.4	**97.1**	1.5

Head-size effect after matched tuning. In the main-text ablations, the small-head row on each benchmark reuses the default hyperparameters of the large head (Tables 7, 11–13). The small-head sweeps above show that part of the larger head-size gaps on CALVIN and WidowX reflects this per-benchmark tuning rather than head capacity.

H.2 An autoregressive action head

To test a family outside the continuous flow-matching head, we replace the head with a discrete autoregressive action head with 256 bins on top of the same SmolVLM-0.5B encoder and run the same representative sweep on LIBERO (Table 19). The AR default uses cosine scheduling and reaches 95.4% average success, 3.2 points below the continuous-head default. The tier ordering is preserved: training dynamics (shuffling off or a learning rate of 5×10^{-4}: $95.4 \rightarrow 0.0$; VLM LR multiplier 1.0: $95.4 \rightarrow 36.9$), then task configuration (largest drop 5.0 points, H=30), then architecture (largest drop 2.9 points, small head). This experiment covers a single benchmark and a single encoder, so whether the hierarchy transfers to autoregressive-tokenization VLAs in general remains open.

H.3 Seed variance

All numbers in the main-text tables were obtained from a single training run (seed 0) and list the best checkpoint of that run. We retrained the default SimVLA with three independent seeds (0, 1, 2) on every benchmark and report mean $\pm$ sample standard deviation. As expected, the three-seed means are slightly lower than the single best-run numbers (for instance LIBERO $98.6 \rightarrow 98.1 \pm 0.5$). For a comparison under the same protocol we also reproduced X-VLA [Zheng et al., 2025b] with three seeds on LIBERO; its average is 97.5 ± 0.4 (Table 20), versus the 98.1 reported in the original paper.

Table 24 repeats the LIBERO ablation table with three seeds for every row. Using the default standard deviation (± 0.5) as the noise scale, the head-scale delta (0.8) and the Florence-2 backbone delta (1.1) are within the seed-noise scale, whereas the injection-scheme deltas (6.8–7.6) and the action-objective deltas (2.9–3.6) are clearly above it. The training-dynamics deltas (shuffling off: 93.7 points, VLM LR multiplier 1.0: 54.3 points) are one to two orders of magnitude larger than the seed standard deviations. The architecture tier therefore remains the smallest measured tier whether its small deltas are treated as real or as noise.

H.4 A third backbone family

The backbone row of Table 7 swaps SmolVLM-0.5B for Florence-2-0.9B and therefore changes size, pretraining corpus, model family and training objective at once. It should be read as a backbone-family swap and not as a controlled scale ablation. We additionally evaluated Qwen3-VL-2B [Qwen Team, 2025] under the default LIBERO configuration (Table 25). Over a $4\times$ range in backbone size (0.5B$\rightarrow$ 0.9B$\rightarrow$ 2B), the backbone-swap deltas stay at 0.9 (Florence-2) and 1.5 (Qwen3-VL-2B) points, much smaller than the training-dynamics maximum.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the attribution hierarchy with quantitative ranges, report benchmark results (single-run and three-seed), describe the contribution as methodological and empirical, and explicitly scope the findings to the simple VLM-encoder plus lightweight-head family (Section 5, Limitations).

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations subsection in Section 5 discusses limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper is empirical and does not include theoretical results or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3.3 details the training recipe (normalization, shuffling, optimization). The Appendix (Training Hyperparameters) provides full hyperparameter tables for all benchmarks. Code is available at https://frontierrobo.github.io/SimVLA/.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at https://frontierrobo.github.io/SimVLA/. All simulation benchmarks (LIBERO, CALVIN, WidowX, Google Robot) use publicly available datasets. The Galaxea Open-World Dataset is described in the Appendix.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details are in Section 3.3 and the Appendix (Training Hyperparameters). Evaluation protocols follow each benchmark's standard setup as described in Section 4.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We follow the standard evaluation protocol of each benchmark (e.g., 500 rollouts for LIBERO, 1000 chains for CALVIN).

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The Appendix (Training Hyperparameters) specifies GPU type (H100), number of GPUs (4 for simulation, 64 for real-robot), batch sizes, and training steps. Table 1 reports peak VRAM usage.

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research involves robotic manipulation in controlled lab and simulation settings. No human subjects, private data, or dual-use concerns are involved.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Broader Impact subsection discusses positive impacts (reproducibility, accessibility, and reduced research cost) and potential risks (safety validation, human oversight, and labor impacts) of stronger robotic manipulation policies.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: SimVLA is a robotic manipulation policy evaluated in controlled settings. The model does not pose risks comparable to generative language or image models.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All benchmarks (LIBERO, CALVIN, WidowX, Google Robot, BEHAVIOR-1K), baseline models, and the SmolVLM, Florence-2, and Qwen3-VL backbones are cited with their original papers.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the SimVLA codebase with documentation at https://frontierrobo.github.io/SimVLA/.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core methodology.