IN-CONTEXT STRUCTURED DEMONSTRATION FOR GENERALIZABLE VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models often degrade sharply in out-of-distribution (OOD) scenarios. In-context imitation learning (ICIL) enables test-time adaptation by retrieving expert demonstrations and appending them as contextual prompts. Existing frameworks, however, underutilize these trajectories: by conditioning on raw observations and continuous actions alone, the policy observes *what* the expert did without the task structure that explains *why*. We propose **StellaVLA**, a framework that advances ICIL by distilling in-context structured rationales into VLA models. First, an offline pipeline extracts a hierarchical rationale from raw expert demonstrations without manual annotation, articulating both high-level semantic sub-goals and fine-grained kinematic movements through a "language-as-action" paradigm. The same rendering applies to real-robot, human-hand and XR-retargeted demos. Second, a parallel dual-training paradigm uses these rationale-augmented trajectories as supervision, training the VLA to predict continuous actions and to articulate the underlying rationale from one shared representation. During deployment, an asymmetric inference strategy strips away the language head, enabling high-frequency control guided by KV-cached expert rationale. Extensive experiments across simulation benchmarks (*e.g.*, LIBERO and VLA-Arena) and real-world manipulation show that StellaVLA effectively exploits heterogeneous demonstrations as structured context, consistently improving both in-distribution performance and OOD robustness, achieving state-of-the-art performance without any robotics pretraining.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a prominent paradigm for robotic manipulation (Kim et al., 2024; 2025; Black et al., 2024). Built upon powerful Vision-Language Models (VLMs), these policies are expected to seamlessly translate visual and textual inputs into physical actions. However, they often exhibit significant fragility and performance collapse when deployed in novel, out-of-distribution (OOD) scenarios (Fang et al., 2025; Zhou et al., 2025), where recovering typically requires collecting new data and fine-tuning. To achieve zero-shot test-time adaptation, In-Context Imitation Learning (ICIL) (Fu et al., 2025; Zhang et al., 2026d; Sridhar et al., 2025) has emerged as a promising technique. By appending retrieved expert trajectories, which typically comprise sequences of visual observations and low-level actions, as contextual prefix prompts, ICIL equips VLA policies with a dynamic non-parametric memory. Drawing upon this retrieved experience, the robot can infer desired task executions on the fly and mimic expert behaviors, thereby adapting to new environments without requiring any updates to its pre-trained weights.

Despite these advancements, existing ICIL frameworks severely underutilize the rich information embedded within expert trajectories. By solely conditioning on raw visual observations and continuous low-level actions, they encourage superficial action mimicry and fail to extract the underlying structured rationale (Zhang et al., 2026d; Jang et al., 2026; Sridhar et al., 2025; Fu et al., 2025). Consequently, the policy observes *what* the expert did without understanding *why*, often treating demonstrations as noise and falling back on its pre-trained priors (Xu et al., 2026b), causing the policy to rely excessively on its pre-trained priors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

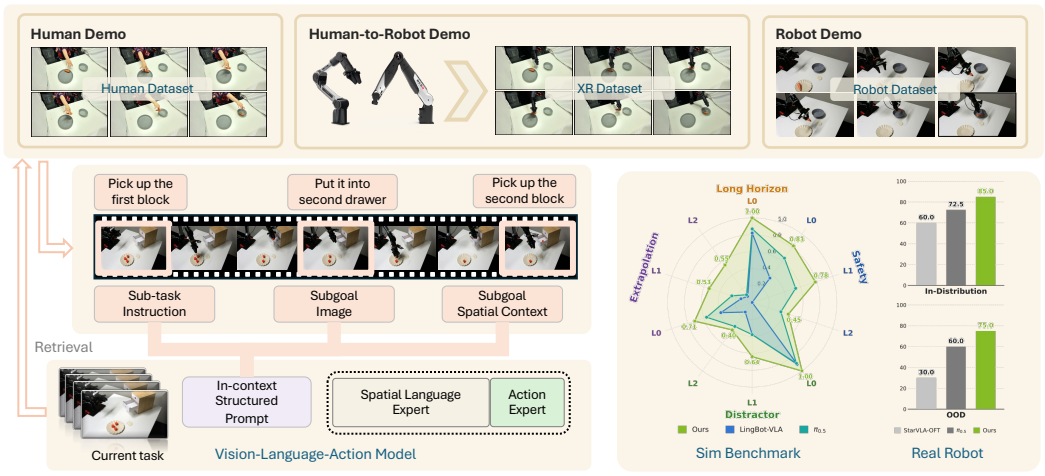


Figure 1: Overview of **in-context structured demonstration**. Human and robot demonstrations are rendered into structured examples of sub-task instructions, visual sub-goals, and spatial context, then retrieved to prompt a VLA model. A spatial-language expert supervises these rationales during training only. *Right*: VLA-Arena and real-robot results (Table 2 and Section 4.3).

Reasoning provides the missing structure. Foundation models generalize better when the intermediate reasoning is made explicit rather than left implicit (Zhong et al., 2026; Zawalski et al., 2024; Mark et al., 2026; Belkhale et al., 2024): decomposing a task into logical, intermediate steps exposes the *why* behind an action rather than just the *what*. This inspires a question: **How can we enable VLA models to leverage the latent thinking process embedded in these retrieved expert trajectories, rather than merely mimicking their actions, to achieve true OOD generalization?**

In this paper, we propose StellaVLA (Figure 1), a novel framework that empowers VLA models by advancing ICIL through in-Context structured distillation. By synergizing explicit language supervision with rationale-augmented context, our approach upgrades the ICIL process from superficial action mimicry to genuine rationale imitation. To achieve this, we first construct a unified cross-embodiment demonstration pool via an automated offline extraction pipeline that translates raw expert trajectories into a structured, rationale-augmented context. Adopting the "language-as-action" paradigm (Lin et al., 2026b; Chen et al., 2026b), a VLM agent segments long-horizon trajectories and generates hierarchical verbal rationales, comprising high-level *semantic* rationales (sub-goals) and fine-grained *kinematic* rationales (movements). These rationales augment each demonstration alongside the original observations and actions at zero human annotation cost.

Unlike prior ICIL methods that rely on prompting alone (Zhang et al., 2026d; Jang et al., 2026), StellaVLA couples in-context prompting with explicit rationale supervision through a parallel dual-training paradigm. During training, the model conditions on retrieved rationale-augmented demonstrations and is jointly optimized to predict both low-level actions and their corresponding rationales, allowing it to internalize not only *what* to do but also *why*. Through this dual objective, expert reasoning is distilled into the policy: language supervision grounds actions in their semantic intent, while retrieved rationales provide structured procedural guidance that can be reused in similar situations. During inference, the spatial-language expert is removed entirely. The policy relies solely on the action expert, conditioned on the KV-cached demonstration prefix, enabling efficient high-frequency control without autoregressive decoding overhead while retaining the procedural knowledge acquired during training.

In summary, our main contributions are:

- We propose StellaVLA, a retrieval-augmented VLA framework that advances ICIL from surface-level action mimicry to structured rationale imitation, enabling more effective use of retrieved demonstrations for robust OOD generalization.
- A parallel dual-training design, fed by a zero-human-cost offline extraction pipeline, that internalizes expert reasoning during training and strips the spatial-language expert at inference, leaving the control loop free of autoregressive decoding overhead.

- Extensive simulation and real-robot manipulation experiments show that StellaVLA reaches an overall VLA-Arena score of 0.68 against 0.56 for the strongest baseline *without any robotics pretraining*, while achieving **98.8%** average success in LIBERO and leading in LIBERO-Plus and our real-robot benchmark.

2 Related Work

In-context adaptation for VLA models. Vision-language-action (VLA) models adapt pretrained vision-language backbones for robotic control through token-based, regression, or flow-matching action prediction, with recent work further improving efficiency and policy representations (Kim et al., 2024; 2025; Black et al., 2024; Physical Intelligence et al., 2025; Pertsch et al., 2025; NVIDIA et al., 2025; Shukor et al., 2025; Hung et al., 2025; StarVLA Community, 2026; Pei et al., 2026; Xu et al., 2026a; Li et al., 2025; Lin et al., 2026a; Zhang et al., 2025b). While these models achieve strong performance, adapting to novel tasks typically requires additional demonstrations and fine-tuning (Wu et al., 2026). Recent deployment-time approaches instead adapt frozen policies by updating fast weights or memory, or by conditioning on retrieved demonstrations and multimodal context (Sun et al., 2020; Jiang et al., 2026; Zhang et al., 2026c; Feng et al., 2026; Patel et al., 2026; Zhang et al., 2026d; Liu et al., 2025; Shi et al., 2026; Fu et al., 2025; Sridhar et al., 2025; Jang et al., 2026). Existing ICIL methods, however, primarily represent demonstrations as observation–action sequences (Chen et al., 2026a; Zhou et al., 2026; Sridhar et al., 2025), leaving the underlying procedural structure implicit. StellaVLA instead retrieves structured demonstrations composed of semantic sub-goals and grounded motion, enabling adaptation through explicit procedural guidance without parameter updates.

Language as action representation. Language has emerged as an effective semantic interface for robot policies, representing low-level actions, sub-tasks, feedback, and action-oriented reasoning while preserving the capabilities of pretrained VLMs (Zha et al., 2026; Hancock et al., 2025; Lin et al., 2026b; Chen et al., 2026b; Shi et al., 2025; Gao et al., 2026b; Zhang et al., 2026b; Zhao et al., 2025; Zhong et al., 2026). Recent embodied chain-of-thought methods explicitly generate intermediate reasoning before action prediction (Zawalski et al., 2024; Lee et al., 2025; Huang et al., 2026), improving generalization at the cost of autoregressive decoding during deployment. StellaVLA instead uses language as a shared representation for both retrieved demonstrations and training supervision. Structured rationales supervise the policy during training but are never decoded at inference, retaining efficient high-frequency control.

Cross-embodiment demonstrations. Human demonstrations and cross-embodiment data broaden the diversity of available experience, but differ substantially in embodiment and action space. Existing approaches bridge this gap through shared latent actions, motion representations, world models, retargeting, or direct video conditioning (Bu et al., 2025; Ye et al., 2024; Bi et al., 2025; Gao et al., 2026a; Chen et al., 2025; Hui et al., 2026; Chen et al., 2026a; Zhou et al., 2026). Rather than learning a shared action space, StellaVLA converts real-robot, human-hand, and XR-retargeted demonstrations (Wang et al., 2026) into a unified structured representation with a common spatial grounding. Off-embodiment demonstrations serve only as contextual guidance, while executable actions are always supervised and predicted in the target robot's native control space.

3 Methodology

In this section, we detail the StellaVLA framework, illustrated in Figure 2, which advances ICIL with structured contextual rationales to improve VLA generalization. We first present an automated offline pipeline that converts raw trajectories into rationale-augmented segments (Section 3.1), then introduce a parallel dual-training paradigm that uses these trajectories as both explicit supervision and in-context prompts (Section 3.2), and finally describe an asymmetric inference strategy that leaves the control loop free of autoregressive decoding overhead (Section 3.3).

3.1 Offline Structured Context Extraction

Raw trajectory formulation. An expert demonstration is recorded as $\tau = \{(o_t, a_t)\}_{t=1}^{T}$, where o_t denotes sensory input (e.g., multi-view RGB and proprioception) and $a_t \in \mathbb{R}^{d_a}$ is the continuous

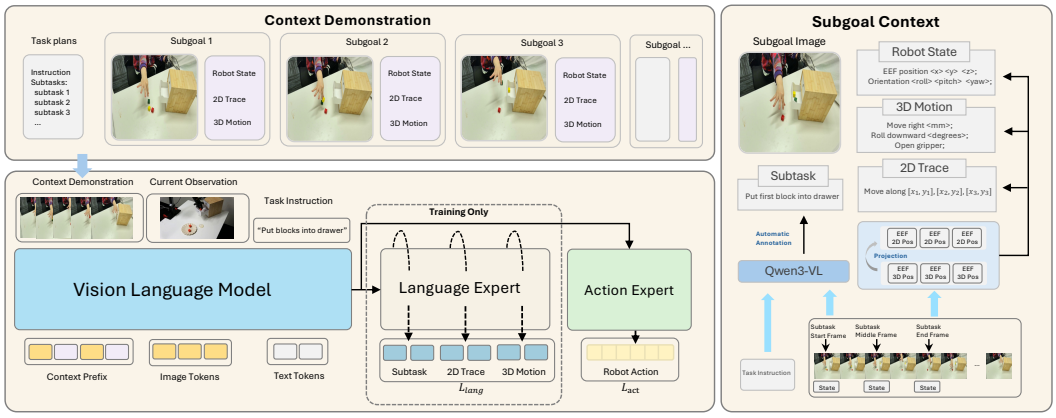


Figure 2: **Overview of StellaVLA.** *Top left:* a retrieved demonstration is a task plan plus sub-goals, each with a keyframe, robot state, 2D trace, and 3D motion. *Right:* these annotations are generated offline from VLM outputs and robot trajectories, without human labeling. *Bottom left:* the backbone encodes the demonstration with the current observation and instruction into one representation, read in parallel by an action expert and, during training only, a spatial-language expert. At inference the latter is dropped, leaving a single forward pass.

control command (e.g., end-effector pose and gripper state). Such trajectories may come from diverse embodiments and collection paradigms, including real-robot teleoperation, human hand tracking, or XR-retargeted demonstrations (Wang et al., 2026); for human-hand recordings, the tracked wrist takes the role of the end effector, and each source's native grounding point is mapped onto the real robot's convention (Appendix C). Although they encapsulate successful executions, the raw numerical actions lack an explicit semantic rationale, which hinders cross-embodiment reasoning.

Semantic segmentation. To bridge the semantic gap without incurring prohibitive human annotation costs, we introduce an automated offline pipeline (Figure 2, right) that infers a sub-goal decomposition from the observed execution. Given a raw trajectory τ and its high-level task instruction $\mathcal{I}$, we use an off-the-shelf vision-language model (Qwen3-VL; Bai et al., 2025) to decompose the continuous trajectory into K discrete, semantically meaningful segments. For each segment $k \in \{1, \ldots, K\}$ spanning time steps $t_{\text{start}}^{(k)}$ to $t_{\text{end}}^{(k)}$, the VLM analyzes the visual changes to identify the high-level sub-goal achieved. We write $k(t)$ for the segment containing step t and s_t for the sub-task label of that step. The decomposition is inferred post hoc from the observed execution and is not manually verified: it is a plausible account of the task rather than the demonstrator's intent.

Kinematic verbalization via language-as-action. Once the trajectory is segmented, we adopt a "language-as-action" paradigm that deterministically translates the physical actions within each segment into structured text. Because VLAs are built upon VLMs with a strong affinity for semantic language (Hancock et al., 2025; Zhang et al., 2026a), these rationales can be processed with the original text vocabulary, without introducing specialized action tokens.

This two-tier process generates a **structured context** l_k for each segment, pairing a *semantic rationale* with a *kinematic rationale*. The semantic rationale is the sub-goal description the VLM identifies from the visual observations, such as "reach for the handle of the blue mug". The kinematic rationale verbalizes fine-grained motion as a 3D movement in the workspace, quantized into millimetres and degrees (e.g., "move right ⟨mm⟩; roll downward ⟨degrees⟩; open gripper"), together with a 2D trace obtained by projecting that trajectory onto the camera plane with known intrinsics and extrinsics. This dual form encourages the model to reason about spatial logic and visual grounding through text tokens rather than emitting raw floating-point actions.

Both kinematic fields are produced by a single deterministic verbalizer Φ, which maps any contiguous span of actions to its 3D displacement in the workspace and to the 2D projection of that displacement onto the image plane. Applying Φ over a whole segment yields the movement description carried inside l_k; we reuse the *same* operator at the much shorter scale of an action chunk in Section 3.2, which places the retrieved demonstration and the policy's own prediction in one

common vocabulary. The sub-goal description, by contrast, is attached to every frame: each step t carries the sub-task s_t that the expert is executing in the observation o_t.

Rationale-augmented demonstration pool. Through this process, the raw trajectory τ is transformed into a rationale-augmented trajectory $\tau_{\text{rat}} = \{(o_t, a_t, s_t, l_{k(t)})\}_{t=1}^{T}$, in which each time step is grounded its observation, action, sub-task label s_t and the *segment-level* structured rationale $l_{k(t)}$. Processing all available expert demonstrations yields the demonstration pool $\mathcal{D}_{\text{pool}} = \{\tau_{\text{rat}}^{(i)}\}_{i=1}^{N}$, into which off-embodiment recordings enter as context only.

3.2 Parallel Dual-Training Paradigm

Retrieval-augmented prompt formulation. During training, the learning process is formulated as a leave-one-out retrieval task. To train the policy to imitate a target trajectory τ_{tgt} sampled from $\mathcal{D}_{\text{pool}}$, the system queries the remaining pool $\mathcal{D}_{\text{pool}} \setminus \{\tau_{\text{tgt}}\}$ and retrieves a rationale-augmented expert trajectory by the language embedding of the task instruction; the querying episode is excluded, so no sample retrieves its own demonstration. Retrieval is therefore purely linguistic, which is what lets a demonstration recorded on another embodiment be retrieved for a robot episode. The retrieved trajectory is formatted into a contextual prefix prompt $\mathcal{P}_{\text{demo}}$ interleaving keyframes and text. The input to the VLA model at time step t concatenates the retrieved demonstration, the current task instruction $\mathcal{I}$, and the current observation o_t:

$$x_t = \big[\mathcal{P}_{\text{demo}} \oplus \mathcal{I} \oplus o_t\big], \tag{1}$$

where $\oplus$ denotes sequence concatenation. Because $\mathcal{P}_{\text{demo}}$ carries the l_k descriptions of the expert, it acts as an in-context learning signal that supplies a template for how to plan in similar scenarios. A demonstration dropout sets $\mathcal{P}_{\text{demo}} = \emptyset$ per sample with probability p, matched to how often a same-task demonstration is retrievable at deployment (Appendix A).

Parallel experts architecture. The core of our policy is a unified vision-language backbone f_θ equipped with two parallel experts (Figure 2, bottom left). The multimodal sequence x_t, closed by H action-placeholder tokens, is processed in a single forward pass into hidden states $h_t = f_\theta(x_t)$, which are read by two experts. The **action expert**, a two-block residual MLP head, maps the final-layer hidden states at the placeholder positions to the continuous action chunk $\hat{A}_t = (\hat{a}_t, \ldots, \hat{a}_{t+H-1})$ spanning the next H control steps. The **spatial-language expert**, the native autoregressive language-model head, emits the rationale $\hat{c}_t = (\hat{s}_t, \hat{m}_t)$ of the current step: the sub-task $\hat{s}_t$ being executed in o_t, and the verbalized 3D and 2D movement $\hat{m}_t$ of that same action chunk. These two experts operate strictly in parallel: neither read-out is an input to the other, they share h_t and nothing else, and the token layout keeps this so by construction, with the placeholders closing the user turn while $\hat{c}_t$ occupies the assistant turn. There is no diffusion loop and no autoregressive decoding of actions.

Spatial-language supervision. While the retrieved demonstration provides segment-level context, spatial-language supervision is defined at each time step as

$$c_t = \big(s_t,\ \Phi(A_t)\big), \tag{2}$$

where s_t is the sub-task label for o_t, $A_t = (a_t, \ldots, a_{t+H-1})$ is the ground-truth action chunk, and $\Phi(A_t)$ verbalizes its 3D and 2D movement. The sub-task label provides semantic supervision for the current task stage; the 2D movement grounds the prediction in o_t, while the 3D movement provides kinematic supervision in the workspace. Both targets are derived automatically from the action labels and the recorded gripper track, so the supervision costs no annotation. The 3D movement and the action label are one physical quantity in two encodings, mutually convertible up to quantization; being quantized, it is too coarse to actuate, which is why it does not replace the regression head.

Dual rationale objective. The model is optimized to minimize the joint objective:

$$\mathcal{L} = \mathcal{L}_{\text{act}}\big(\hat{A}_t, A_t\big) + \lambda\, \mathcal{L}_{\text{lang}}\big(\hat{c}_t, c_t\big), \tag{3}$$

where $\mathcal{L}_{\text{act}}$ is the L_1 regression loss on the continuous control commands, $\mathcal{L}_{\text{lang}}$ is the autoregressive cross-entropy loss on the structured rationale, and λ balances the two terms. We set $\lambda = 0.3$ in all experiments. Section 4.4 reports how this choice trades in-distribution precision against robustness. Since Φ is deterministic, A_t and $\Phi(A_t)$ describe the same motion in continuous and linguistic form. The auxiliary term therefore shapes the shared representation during training and has no path into the action head: its cost is confined to training and its effect to the representation. Section 4.4 reports what it buys and at what price.

3.3 Asymmetric Inference and Caching

Action-only execution. A central limitation of embodied chain-of-thought paradigms is the latency of autoregressive text generation at inference, which conflicts with the timing requirements of high-frequency control (Zhao et al., 2025; Zhong et al., 2026; Zawalski et al., 2024). Because the action expert and the spatial-language expert read h_t in parallel, and the whole effect of the language target is exerted on h_t during training, the action read-out is complete without the language read-out. At inference the latter is simply not executed: an action is one forward pass through the backbone and one MLP, with no token generation anywhere in the control loop. The cost so avoided is not marginal; decoding the rationale at every step raises per-step latency by roughly 3.1 s, about $36\times$ the action-only path measured in the same run (Table 9), which at the real robot's 30 Hz rate exceeds the 0.53 s spanned by a 16-step chunk roughly sixfold.

Demonstration prefix caching. The retrieved prefix $\mathcal{P}_{\text{demo}}$ is pinned when a rollout begins and is immutable thereafter. Because the backbone is causal and the prefix precedes all step-dependent tokens, its key–value (KV) cache is computed once at the first step and reused, so every later step forwards only the live suffix: the current images, the current state, the instruction, and the placeholders. Caching an image-and-text demonstration halves per-step latency, although it does not reduce the cost of conditioning to zero (Table 9). Appendix A reports the size of the prefix and verifies that the reuse is exact.

4 Experiments

Our experiments ask what a policy gains from reading a demonstration as *structure*, that is, a task plan, sub-goal descriptions, and verbalized kinematics, rather than as a stream of pixels and coordinates. Three questions organize the evaluation: (i) where its benefit concentrates as the test distribution moves away from training, (ii) whether the structured demonstration is actually used rather than ignored, and (iii) which part of the demonstration carries that benefit.

4.1 Experimental Setup

StellaVLA couples a Qwen3-VL-4B vision-language backbone with an action expert. At each step the model reads a third-person and a wrist RGB image, a language rendering of the robot state, the task instruction, and one retrieved same-task structured demonstration. All models are fully fine-tuned from Qwen3-VL-4B-Instruct for 30k steps at a global batch size of 128 on $4\times$H200 GPUs. A context-demonstration dropout matches demonstration availability at deployment: 0.0 where a same-task demonstration is always retrievable (LIBERO, LIBERO-Plus) and 0.5 otherwise (VLA-Arena, the real robot). We report task success rate under each benchmark's official protocol and compare against StarVLA-OFT, a matched control trained on the same data with the same backbone and action expert but without a retrieved demonstration or spatial-language supervision. Appendix A gives the remaining hyperparameters and the protocol.

Table 1: Zero-shot robustness on LIBERO-Plus (Fei et al., 2025). All models are trained on standard LIBERO and tested on the perturbed tasks without retraining. *Avg.* is the *task-count-weighted* mean over the seven perturbation categories, excluding *Orig.* Best per column in **bold**.

Method	Orig.	Cam.	Robot	Noise	Layout	Backg.	Light	Lang.	Avg.
OpenVLA	76.5	1.1	4.1	19.3	31.6	25.3	4.4	26.8	16.0
OpenVLA-OFT	97.1	59.7	37.2	76.7	77.1	92.4	85.8	81.5	71.4
π_0	94.2	15.8	6.6	79.4	70.4	78.5	79.6	61.0	53.8
π_0-FAST	85.5	66.4	24.8	75.8	70.3	67.7	73.0	63.3	62.5
NORA	87.9	4.0	41.1	17.6	63.9	50.5	31.0	67.0	38.7
WorldVLA	79.1	0.3	30.2	12.2	39.4	14.5	29.4	44.2	24.3
UniVLA	95.2	4.3	50.3	25.3	34.3	80.0	59.1	71.8	44.0
RIPT-VLA	97.5	58.3	36.7	73.8	76.5	90.4	87.9	80.1	70.4
StarVLA-OFT	96.6	47.0	60.1	73.1	79.2	**95.3**	**96.3**	87.0	75.0
StellaVLA	**98.8**	**70.5**	**74.8**	**92.8**	**79.3**	95.2	95.7	**95.3**	**85.1**

Table 2: Success rate on VLA-Arena across 3 difficulty levels: L0 (in-distribution), L1 (intermediate generalization), and L2 (hardest). Rows are the 11 task suites grouped into four categories (Safety, Distractor, Extrapolation, Long Horizon). Baseline results are from the VLA-Arena leaderboard (Zhang et al., 2025a). †StellaVLA is trained *without robotics pretraining*. Best per column in **bold**.

Task Suite	π_0			$\pi_{0.5}$			Motus			GR00T-N1.7			LingBot-VLA 2.0			DM0.5			**StellaVLA**†		
	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2	L0	L1	L2
Safety																					
Static Obst.	0.98	0.74	0.32	0.90	0.64	0.50	0.64	0.67	0.41	0.93	0.85	0.69	0.99	0.72	0.57	0.99	0.61	0.59	**1.00**	**1.00**	**1.00**
Cautious Grasp	0.84	0.08	0.00	0.38	0.16	0.00	0.62	0.25	0.11	0.91	0.34	0.12	0.96	0.23	**0.25**	0.90	0.19	0.13	**1.00**	**0.46**	0.08
Hazard Avoid.	0.74	0.00	0.00	0.62	0.40	0.28	0.65	0.34	0.21	0.61	0.31	0.24	0.87	0.00	0.28	**0.89**	0.39	0.42	0.66	**0.94**	**0.70**
State Pres.	0.98	0.64	0.48	0.80	0.80	0.74	0.63	0.63	0.72	0.95	**0.96**	**0.95**	0.93	0.92	0.70	**0.99**	0.71	0.71	0.90	0.90	0.76
Dynamic Obst.	0.92	0.64	0.10	0.54	0.70	0.26	0.58	0.33	0.11	0.85	0.71	0.25	**0.97**	0.79	0.37	**0.97**	**0.83**	0.30	0.80	0.62	**0.40**
Distractor																					
Static Distr.	0.92	0.02	0.02	0.90	0.04	0.14	0.69	0.19	0.13	0.94	0.17	0.07	0.99	0.23	0.21	0.98	0.29	0.19	**1.00**	**0.56**	**0.40**
Dynamic Distr.	0.78	0.70	0.18	0.88	0.72	0.56	0.77	0.51	0.26	0.88	0.67	0.33	0.99	**0.78**	0.67	0.97	**0.78**	0.69	**1.00**	0.74	**0.78**
Extrapolation																					
Prep. Comb.	**0.76**	0.10	0.00	0.54	0.04	0.00	0.41	0.07	**0.01**	0.73	0.03	**0.01**	0.66	0.03	0.00	0.72	**0.21**	**0.01**	0.64	0.14	0.00
Task Workflows	0.72	0.00	0.00	0.52	0.22	0.26	0.38	0.25	0.14	**0.75**	0.26	0.21	0.63	0.20	0.22	0.57	0.31	0.50	0.64	**0.72**	**0.84**
Unseen Obj.	0.80	0.52	0.04	0.64	0.50	0.08	0.63	0.65	0.19	0.71	0.63	0.41	0.69	0.71	0.42	0.95	0.76	0.30	**1.00**	**1.00**	**0.80**
Long Horizon																					
Long Horizon	0.92	0.02	0.00	0.87	0.00	0.00	0.61	0.04	**0.01**	0.79	**0.07**	0.00	0.98	0.00	0.00	0.78	0.00	0.00	**1.00**	0.00	0.00
Mean	0.85	0.31	0.10	0.69	0.38	0.26	0.60	0.36	0.21	0.82	0.45	0.30	**0.88**	0.42	0.34	**0.88**	0.46	0.35	**0.88**	**0.64**	**0.52**
Overall	0.42			0.44			0.39			0.52			0.54			0.56			**0.68**		

4.2 Simulation Benchmarks

We first evaluate StellaVLA on three simulation benchmarks (Figure 4, Appendix B), probing in-distribution competence, task-level generalization, and zero-shot robustness.

In-Distribution Validation in LIBERO. StellaVLA is on par with the strongest recent VLA policies (Table 4, Appendix B), separated from them by less than one standard error of the rollout estimate, so conditioning on structured demonstrations costs nothing in distribution. Against the demonstration-free StarVLA-OFT the margin is negligible on Object but several points on Goal and Long, which is where the current frame underdetermines the intended outcome or the order of sub-goals: even in distribution the demonstration is redundant as perception and useful as a plan.

Generalization Evaluation on VLA-Arena. VLA-Arena (Zhang et al., 2025a) evaluates task-level generalization across 11 suites at three difficulty levels, from L0 (in-distribution) to L2 (hardest), with training restricted to L0 data. We follow the official protocol and compare against the current leaderboard entries π_0, $\pi_{0.5}$, Motus, GR00T-N1.7, LingBot-VLA 2.0, and DM0.5 (Dexmal Team, 2026). At test time, StellaVLA receives one structured demonstration of the target task without any parameter update. Demonstrations are always retrieved from the L0 training set at every difficulty level; the leaderboard baselines receive only the instruction, so the comparison measures the method together with its access to this additional context rather than the method alone.

The shape of the curve matters more than its height (Table 2). In distribution StellaVLA and the strongest baseline, DM0.5, are at the same level; the gap opens only once the task varies, by $+0.18$ at both L1 and L2. This is what the method predicts: a demonstration re-supplies the task at test time rather than improving what is already learned at L0. Extrapolation makes the mechanism concrete, since StellaVLA holds its Unseen Objects score across all three levels, where a novel object changes the referent but not the procedure. Long Horizon bounds the claim instead: every method collapses at L1 and L2, so a prefix that is fixed for the episode cannot re-plan once execution has drifted.

Zero-Shot Robustness on LIBERO-Plus. LIBERO-Plus (Fei et al., 2025) evaluates zero-shot robustness by perturbing LIBERO tasks along seven axes, including viewpoint, robot state, sensor noise, object layout, background, lighting, and language. We directly evaluate the LIBERO-trained checkpoint without retraining and compare with the zero-shot baselines reported there: OpenVLA (Kim et al., 2024), OpenVLA-OFT, π_0, π_0-FAST (Pertsch et al., 2025), NORA (Hung et al., 2025), WorldVLA (Cen et al., 2025), UniVLA (Bu et al., 2025), and RIPT-VLA (Tan et al., 2025). StellaVLA averages 85.1%, ten points above the matched StarVLA-OFT control (Table 1).

The seven axes are divided by what they corrupt. Four of them degrade the observation or the wording while leaving the task unchanged, and this is where the gains concentrate: camera viewpoint, sensor noise, robot initial state, and language. A structured demonstration is stated in language and

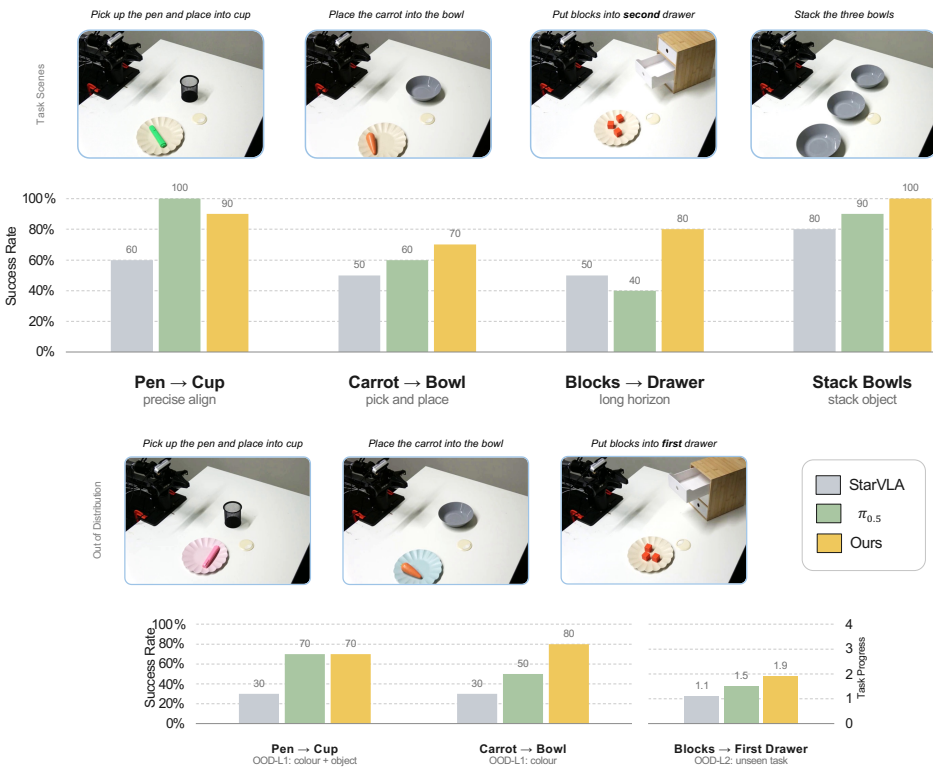


Figure 3: Real-robot evaluation on the AgileX Piper, 10 rollouts per cell, with each evaluation scene shown above the corresponding success rates. *Top*: in-distribution success rate on the four tasks. *Bottom left*: *OOD-L1*, which perturbs object attributes and is comparable within a task rather than across tasks. *Bottom right*: *OOD-L2*, an unseen task placing the blocks in the first drawer, scored by task progress out of four.

in a canonically grounded image-plane path, so it survives a moved camera or a reworded instruction, each of which invalidates a learned observation–action correspondence. On the language axis the retrieval query is the perturbed instruction itself, so no privileged task identity enters the comparison. Object layout is the informative exception, with no gain at all: it is the one axis that moves the objects themselves, so the demonstration's verbalized kinematics describe an arrangement that no longer holds and the policy must fall back on the live observation. What transfers is a procedure, not a trajectory. Background and lighting are near saturation for both models, where perception has lost nothing for the demonstration to restore.

4.3 Real-World Experiments

We evaluate StellaVLA on a 6-DOF AgileX Piper arm with third-person and wrist-mounted RGB cameras, on four tabletop tasks covering precision pick-and-place, articulated-object manipulation, and multi-stage execution. Each task is trained from around 30 expert trajectories, reflecting a low-data regime. Demonstrations are drawn at random from a merged pool of real-robot, human-hand, and XR-retargeted trajectories, with off-embodiment data entering the retrieval pool only and never supplying an action target. Appendix C details the data collection, baselines, and demonstration pool, and reports the three sources to be interchangeable under matched observations.

StellaVLA consistently achieves the strongest performance across both in-distribution and OOD evaluations (Figure 3). Compared with $\pi_{0.5}$, which benefits from large-scale robot pretraining, StellaVLA demonstrates greater robustness under distribution shift, suggesting that structured demonstrations provide complementary procedural guidance beyond low-level action imitation. While the results are averaged over only 20 rollouts per setting and should be interpreted accordingly, the same trend is observed across tasks. None of the evaluated methods solves the unseen OOD-L2 task, though StellaVLA attains the highest intermediate progress (Appendix C), indicating improved procedural generalization without yet achieving zero-shot completion.

Table 3: Results when applying interventions on the in-context structured demo. The first block changes the demonstration given to a fixed checkpoint at evaluation; the last two change its modality at evaluation and at training. Image+Text is the default. LP is the LIBERO-Plus average.

Demonstration	Sp.	Obj.	Goal	Long	AVG	LP
Content, at evaluation						
Correct (Image+Text)	99.6	99.0	99.6	96.8	98.8	85.1
None	72.2	94.4	24.8	58.2	62.4	–
Wrong task	72.6	52.0	0.0	55.0	44.9	–
Modality, at evaluation						
Text-only	99.0	99.8	99.8	96.6	98.8	84.4
Image-only	96.4	98.6	96.0	80.4	92.9	75.7
Modality, at training						
Text-only	98.6	98.6	99.2	92.8	97.3	85.0
Image-only	99.4	99.4	98.8	95.8	98.4	78.7

4.4 Ablation Studies

Role of the demonstration. Serving the same checkpoint with the correct demonstration, with none, and with one from a different task (Table 3) shows that a mismatched demonstration is *worse* than none. This is the informative part: a policy exhibiting behavioral inertia would ignore the context and land at the no-demonstration score whatever it contained. The per-suite pattern locates the effect. Goal collapses, since its tasks share objects and differ only in the intended outcome, so a wrong demonstration overrides the correct instruction that remains in the input; Spatial is unaffected, because the scene itself still specifies where to act. The demonstration is read as a task specification, and it is decisive exactly where the observation underdetermines one.

Demonstration modality. At evaluation time text-only demonstrations nearly match image+text while image-only lags, worst on Long, so what the policy extracts is carried by the *structured language* rather than by the demonstration frames. This predicts the cross-embodiment interchangeability measured on hardware: if what transfers is a verbalized procedure, the embodiment that produced it is not part of the signal.

Training-time modality runs the other way, trading a higher in-distribution average for markedly weaker robustness. Image demonstrations offer an appearance-correspondence shortcut that is available in distribution and gone under perturbation; structured language withholds it and pushes the policy onto the procedural description, which is what survives the shift.

Auxiliary language loss and efficiency. Table 8 (Appendix D) studies the weight λ of the auxiliary language loss. The results indicate a trade-off between in-distribution accuracy and OOD robustness: moderate language supervision improves in-distribution performance, while excessive weighting reduces robustness. We therefore adopt $\lambda = 0.3$, which provides the best overall balance across benchmarks. During inference, the retrieved demonstration prefix is processed once and reused through KV caching, reducing per-step latency from 183 to 91 ms. Moreover, removing the language branch avoids autoregressive decoding, reducing inference latency from 3177 to 88 ms while preserving the procedural knowledge distilled during training. Additional ablations on rationale structure, sub-goal granularity, and latency are provided in Appendix D.

5 Conclusion

We presented StellaVLA, a framework that improves VLA generalization by converting retrieved expert demonstrations into structured in-context guidance. An automated offline pipeline extracts hierarchical semantic and kinematic rationales from heterogeneous demonstrations, while a parallel dual-training objective jointly learns continuous control and spatial-language reasoning. At inference, the spatial-language expert is removed and the fixed demonstration prefix is KV-cached, allowing the policy to benefit from structured context without autoregressive language-decoding overhead. Experiments on LIBERO, VLA-Arena, LIBERO-Plus, and real-world manipulation show that StellaVLA preserves strong in-distribution performance while improving robustness to task and environment shifts, including when demonstrations originate from different embodiments.

AI USE STATEMENT

Generative AI plays two distinct roles in this work. First, as a component of the proposed method, the pretrained vision-language model Qwen3-VL is used offline to segment demonstration trajectories and generate sub-goal descriptions (Section 3.1); these annotations are produced automatically. Second, as a research-assistance tool, we used LLM-based assistants to edit the manuscript for readability, to adapt it to the conference template, to review it for claims not supported by the reported results and suggest more cautious interpretations, and to organize and format the bibliography. All suggested interpretations were checked by the authors against the experimental data, and all bibliographic entries were checked against their sources. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human participants were involved only as demonstrators and evaluators: operators teleoperated the robot, recorded hand demonstrations in XR, and scored real-robot rollouts. The recordings shown in this paper capture hands and tabletop scenes and contain no faces. All operators gave informed consent for their recordings to be collected and used in this research. Real-robot experiments used a filtered control stack with a per-step joint-jump guard to limit unsafe motions (Appendix C). The simulation benchmarks are publicly available. We are not aware of further ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

Section 3 specifies the construction of structured demonstrations, the training objective, and the inference procedure. Appendix A lists the optimization hyperparameters, regularization, retrieval procedure, the design of the action and spatial-language experts, and per-platform input and action spaces. Appendix C describes the real-robot control stack, data collection, construction of the cross-source demonstration pool, success criteria, and the progress metric. All simulation benchmarks are public and evaluated under their official protocols, and the source of every baseline result is stated in the corresponding table. The source code is provided as supplementary material.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

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

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. *arXiv preprint arXiv:2512.13030*, 2025.

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.

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.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 2527–2530, 2012.

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.

Guangyan Chen, Meiling Wang, Qi Shao, Zichen Zhou, Weixin Mao, Te Cui, Minzhao Zhu, Yinan Deng, Luojie Yang, Zhanqi Zhang, Yi Yang, Hua Chen, and Yufeng Yue. See Once, Then Act: Vision-Language-Action Model with Task Learning from One-Shot Video Demonstrations. *arXiv preprint arXiv:2512.07582*, 2025.

Guangyan Chen, Meiling Wang, Te Cui, Zichen Zhou, Qi Shao, Shalfun Li, Hang Su, Roy Gan, Hao Wang, Mengyin Fu, et al. Robots acquire manipulation skills in seconds from a single human video. *arXiv preprint arXiv:2607.20033*, 2026a.

William Chen, Jagdeep Singh Bhatia, Catherine Glossop, Nikhil Mathihalli, Ria Doshi, Andy Tang, Danny Driess, Karl Pertsch, and Sergey Levine. Steerable Vision-Language-Action Policies for Embodied Reasoning and Hierarchical Control. *arXiv preprint arXiv:2602.13193*, 2026b.

Dexmal Team. DM0.5: An open-world foundation model for general-purpose embodied intelligence, July 2026. URL https://www.dexmal.com/blog/dm0.5/index_en.html.

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models. *arXiv preprint arXiv:2510.13626*, 2025.

Yusen Feng, Bingchen Han, Jiangran Lyu, Kai Liu, Yixin Zheng, Yuxuan Wan, Weiheng Liu, Sun Han, Ruiqin Li, Yulong Zhang, Fangfu Liu, Xuesong Shi, Libin Liu, Yizhou Wang, Zhizheng Zhang, and He Wang. WAM-TTT: Steering World-Action Models by Watching Human Play at Test Time. *arXiv preprint arXiv:2607.06988*, 2026.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. Icrt: In-context imitation learning via next-token prediction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Seonghyeon Ye, Qianli Ma, Ruijie Zheng, Pieter Abbeel, Yuke Zhu, Joel Jang, Linxi Fan, et al. DreamDojo: A Generalist Robot World Model from Large-Scale Human Videos. *arXiv preprint arXiv:2602.06949*, 2026a.

Tian Gao, Celine Tan, Catherine Glossop, Timothy Gao, Jiankai Sun, Kyle Stachowicz, Shirley Wu, Oier Mees, Dorsa Sadigh, Sergey Levine, and Chelsea Finn. SteerVLA: Steering Vision-Language-Action Models in Long-Tail Driving Scenarios. *arXiv preprint arXiv:2602.08440*, 2026b.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as Language: Fine-Tuning VLMs into VLAs Without Catastrophic Forgetting. *arXiv preprint arXiv:2509.22195*, 2025.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *Advances in Neural Information Processing Systems*, 38:82782–82802, 2026.

Chenyu Hui, Xiaodi Huang, Siyu Xu, Yunke Wang, Shan You, Fei Wang, Tao Huang, and Chang Xu. Seeing realism from simulation: Efficient video transfer for vision-language-action data augmentation. *arXiv preprint arXiv:2605.02757*, 2026.

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.

Sanghwan Jang, Minjin Jeon, Minsoo Kim, Seong Jin Choi, Dongha Kim, and Hwanjo Yu. Ra-vla: Retrieval-augmented vla for test-time adaptation. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR, 2026. URL https://openreview.net/forum?id=ut6HebnnQe.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Li Fei-Fei, Yuke Zhu, et al. Robottt: Context scaling for robot policies. *arXiv preprint arXiv:2607.15275*, 2026.

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.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. CogVLA: Cognition-Aligned Vision-Language-Action Model via Instruction-Driven Routing & Sparsification. *arXiv preprint arXiv:2508.21046*, 2025.

Tao Lin, Yuxin Du, Jiting Liu, Nuobei Zhu, Yunhe Li, Yuqian Fu, Yinxinyu Chen, Hongyi Cai, Zewei Ye, Bing Cheng, Kai Ye, Yiran Mao, Yilei Zhong, MingKang Dong, Junchi Yan, Gen Li, and Bo Zhao. Evo-Depth: A Lightweight Depth-Enhanced Vision-Language-Action Model. *arXiv preprint arXiv:2605.14950*, 2026a.

Tao Lin, Yuxin Du, Yiran Mao, Zewei Ye, Yilei Zhong, Bing Cheng, Yiming Wang, Jiting Liu, Yang Tian, Junchi Yan, Feiran Wu, Zenan Meng, Hu Wei, Yuqian Fu, Gen Li, and Bo Zhao. LA4VLA: Learning to Act without Seeing via Language-Action Pretraining. *arXiv preprint arXiv:2606.27295*, 2026b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*, 2023.

Min Liu, Deepak Pathak, and Ananye Agarwal. Locoformer: Generalist locomotion via long-context adaptation. *arXiv preprint arXiv:2509.23745*, 2025.

Max Sobol Mark, Jacky Liang, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Dhruv Shah, and Aviral Kumar. Bpp: Long-context robot imitation learning by focusing on key history frames. *arXiv preprint arXiv:2602.15010*, 2026.

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. *arXiv preprint arXiv:2503.14734*, 2025.

Austin Patel, Ben Pekarek, Joel Enrique Castro Hernandez, and Shuran Song. Behavior Prompting Policy: Demonstrations as Prompts for Manipulation. *arXiv preprint arXiv:2606.30457*, 2026.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 10832–10851, 2026.

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.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, 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.

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*, 2026.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models. *arXiv preprint arXiv:2502.19417*, 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.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. Ricl: Adding in-context adaptability to pre-trained vision-language-action models. *arXiv preprint arXiv:2508.02062*, 2025.

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing. *arXiv preprint arXiv:2604.05014*, 2026.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. *arXiv preprint arXiv:1909.13231*, 2020.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

Jen-Wei Wang, Sarthak Kaingade, Andrea Tagliabue, and Nicholas Morozovsky. X-op: Cross-morphology whole-body teleoperation via mpc retargeting. *arXiv preprint arXiv:2606.07934*, 2026.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From Foundation to Application: Improving VLA Models in Practice. *arXiv preprint arXiv:2607.06403*, 2026.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. AVA-VLA: Improving Vision-Language-Action Models with Active Visual Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13453–13463, 2026.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. *Advances in Neural Information Processing Systems*, 38:164448–164473, 2026a.

Siyu Xu, Zijian Wang, Yunke Wang, Chenghao Xia, Tao Huang, and Chang Xu. Affordance field intervention: Enabling vlas to escape memory traps in robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37206–37215, 2026b.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent Action Pretraining from Videos. *arXiv preprint arXiv:2410.11758*, 2024.

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

Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. LAP: Language-Action Pre-Training Enables Zero-Shot Cross-Embodiment Transfer. *arXiv preprint arXiv:2602.10556*, 2026.

Borong Zhang, Jiahao Li, Jiachen Shen, Yishuai Cai, Yuhao Zhang, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. VLA-Arena: An Open-Source Framework for Benchmarking Vision-Language-Action Models. *arXiv preprint arXiv:2512.22539*, 2025a.

Fengnian Zhang, Tao Huang, Siyu Xu, Zhong Jin, and Chang Xu. Revisiting parameter redundancy in vision-language-action models: Insights from vlm-to-vla adaptation. *arXiv preprint arXiv:2606.31382*, 2026a.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. VLM4VLA: Revisiting Vision-Language-Models in Vision-Language-Action Models. *arXiv preprint arXiv:2601.03309*, 2026b.

Wenbo Zhang, Jianxiong Li, Shuai Yang, Sijin Chen, Jiajun Liu, Lingqiao Liu, and Xiao Ma. TTT-VLA: Test-Time Latent Prompt Optimization for Vision-Language-Action Models. *arXiv preprint arXiv:2606.03127*, 2026c.

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*, 2025b.

Yue Zhang, Rui Wang, Jiehong Lin, Zhongrui Wang, and Xiaojuan Qi. Retrieval-VLA: Training-Free In-Context Adaptation for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1358–1367, 2026d.

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. *arXiv preprint arXiv:2503.22020*, 2025.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. ACoT-VLA: Action Chain-of-Thought for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8152–8162, 2026.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, et al. Zero-wam: In-context world-action modeling from human videos for open-ended task generalization. *arXiv preprint arXiv:2608.26103*, 2026.

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.

A Implementation Details

Optimization. All models are fully fine-tuned from Qwen3-VL-4B-Instruct with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-8}), a backbone learning rate of 5×10^{-6} and an action-expert learning rate of 10^{-4} under a cosine schedule (500 warmup steps), gradient clipping 1.0, bf16 mixed precision, and DeepSpeed ZeRO-2, for 30k steps at a global batch size of 128.

Regularization. Two per-sample dropouts regularize training. A 2D gripper-path dropout (0.5) keeps the 3D steering command usable when the pixel trace is missing, so that neither the policy's output nor its conditioning comes to require a pixel path that a given source may not supply. A context-demonstration dropout matches demonstration availability at deployment: 0.0 when a same-task demonstration is always retrievable (LIBERO, LIBERO-Plus) and 0.5 otherwise (VLA-Arena, the real robot). At 0.5 the demonstration-free condition lies in distribution, so withholding the demonstration is a control within a single checkpoint; at 0.0 it is not, which is a caveat on the no-demonstration row of Table 3.

Retrieval. Demonstrations are retrieved by the language embedding of the task instruction alone; no visual features enter the query. In the closed instruction sets used here this reduces to an exact task match, and one of the matching demonstrations is drawn uniformly, with the querying episode excluded so that no training sample retrieves its own demonstration. A fresh element is drawn per sample during training; at evaluation one is pinned when the rollout begins and held for its duration, determined by task and source alone, which makes a change of source a controlled substitution rather than a resampling. We keep retrieval deliberately simple to separate what a demonstration contributes from how well one can be selected.

Action expert and spatial-language expert. A set of action-placeholder tokens is appended to the user turn, causally preceding the language-action chain-of-thought. After a single backbone forward, their final-layer hidden states pass through a two-block residual MLP that regresses the action chunk under an L_1 objective. The compact "sub-task + 3D/2D steering command" chain-of-thought is retained from a language-action formulation (Zha et al., 2026) and supervised against offline-generated annotations by cross-entropy; it is not decoded at action inference. The steering command is the target $\Phi(A_t)$ of Eq. 2, that is, the same action chunk the MLP regresses, rendered by the verbalizer Φ over the chunk span rather than over a whole segment; the sub-task is the per-frame label s_t.

Per-platform instantiation. In simulation the policy reads 256×256 images, predicts end-effector deltas over an action chunk of length 8, and the retrieved demonstration carries up to $K = 10$ sub-goal keyframes. On the real robot it reads 640×480 images, predicts a 7-dimensional chunk of length 16 (six joint-position deltas relative to the episode start plus an absolute gripper width, all q_{99}-normalized), and the demonstration carries up to $K = 8$ keyframes. The real-robot state is rendered in language as the end-effector pose together with the joint angles, in millimetres, degrees, and gripper opening as a percentage.

Protocol and matched control. We report task success rate throughout, following each benchmark's official evaluation protocol, and introduce each comparison set alongside the benchmark it is reported on. One reference recurs everywhere. Beside StellaVLA, the context-demonstration policy, we train *StarVLA-OFT* (StarVLA Community, 2026) with the same backbone, action expert, and training data, differing only in receiving no retrieved demonstration and no auxiliary spatial-language expert. Because StellaVLA conditions on a demonstration while the published baselines do not, this matched control and the with/no/wrong-demonstration intervention of Table 3 carry the claim, rather than the comparison against leaderboard entries. Each configuration is trained once with a fixed seed and evaluated over multiple trials; for reference, the binomial standard error of a LIBERO four-suite average (2,000 rollouts) is about 0.25 points at 98.8% success, which reflects rollout noise only.

Demonstration prefix caching. The retrieved demonstration accounts for roughly 85% of the prompt's tokens, which is what makes caching it worthwhile. The reuse is exact rather than approximate: the largest deviation we measure between actions predicted with and without the cache is about 1.1×10^{-3} in normalized units, so the latency measurements of Appendix D are not confounded by a cache that alters the policy's output.

B Simulation Benchmarks and In-Distribution Results

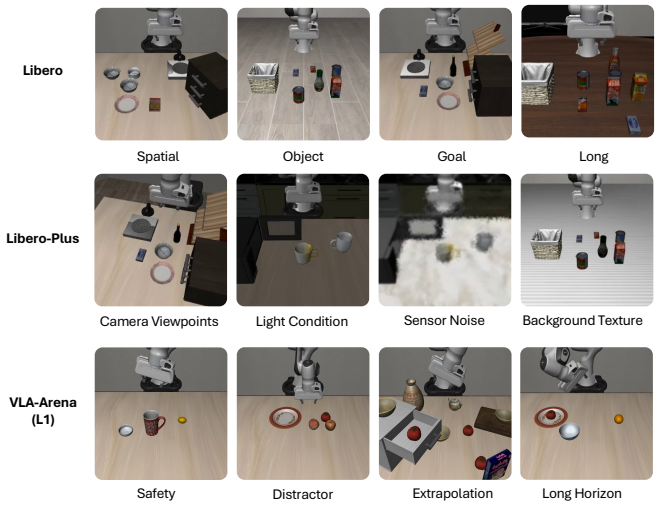


Figure 4: **The three simulation benchmarks.** *Top*: the four standard LIBERO suites. *Middle*: LIBERO-Plus perturbation axes, e.g., camera viewpoint, lighting, sensor noise, and background texture. *Bottom*: VLA-Arena, which changes the task, adding safety constraints, distractors, extrapolation to unseen objects, and long-horizon composition.

Table 4: In-distribution success rate (%) on the standard LIBERO benchmark (Liu et al., 2023). Each suite is evaluated over 500 rollouts; baseline numbers are as reported in the original papers, except our matched demonstration-free control StarVLA-OFT (StarVLA Community, 2026). Best per column in **bold**.

Method	Spatial	Object	Goal	Long	Avg.
MemoryVLA (Shi et al., 2026)	98.4	98.4	96.4	93.4	96.7
ACoT-VLA (Zhong et al., 2026)	99.4	**99.6**	98.8	96.0	98.5
AVA-VLA (Xiao et al., 2026)	99.2	**99.6**	97.9	96.2	98.2
StarVLA-OFT (matched control)	97.8	98.6	96.2	93.8	96.6
CogVLA (Li et al., 2025)	98.6	98.8	96.6	95.4	97.4
Retrieval-VLA (Zhang et al., 2026d)	97.4	98.8	96.3	89.5	95.5
DreamVLA (Zhang et al., 2025b)	97.5	94.0	89.5	89.5	92.6
StellaVLA (Ours)	**99.6**	99.0	**99.6**	**96.8**	**98.8**

C Real-World Setup

Control stack. The AgileX Piper is driven over CAN at 30 Hz without a ROS stack. Each 16-step chunk spans ≈ 0.53 s and is executed in a synchronous receding-horizon loop with a horizon of 8. Joint targets pass through a One Euro filter (Casiez et al., 2012) and a 25° per-step jump guard before reaching the arm. Because the demonstration prefix is fixed for the duration of a rollout, its key–value cache is computed once and reused, and steady-state inference settles at ≈ 205 ms per chunk, within the 0.27 s spanned by the 8 executed steps.

Data collection. The 125 teleoperated episodes (71,702 frames) were collected by leader–follower teleoperation across the four tasks: *pick the pen and put it in the cup*, *pick the carrot and put it in the bowl*, *put the blocks into the second drawer and close it*, and *stack the three bowls on top of each other*. They cover precision alignment, free-space pick-and-place, a long-horizon sequence involving an articulated object, and multi-stage stacking respectively.

Cross-source pool construction. Twenty-six human-hand takes recorded in XR yield two aligned datasets of 26 episodes and 10,996 frames each: one preserving the operator's bare hand, one show-

ing a retargeted Piper arm executing the identical trajectory. The two share episode structure and timing frame for frame, so the appearance of the embodiment is the only variable separating them. The grounding point of the 2D gripper path uses the operator's wrist for the human take and the flange for the retargeted arm, both matched to the real robot's convention, so that a rendered coordinate denotes the same physical quantity regardless of provenance. The three sources are merged by exact task-string match, retrieval excludes the current episode, and the off-embodiment data enters *only* the retrieval pool: no human or retargeted frame ever supplies an action target, so the policy's motor supervision remains entirely real-robot.

Evaluation conditions. Every reported cell is 10 rollouts scored by a human operator against a fixed success criterion, with object placements re-randomized per rollout and held identical across methods. OOD-L1 severity is not calibrated across tasks: it compounds a colour and an object substitution for pen→cup but changes only colour for carrot→bowl, so L1 is comparable within a task rather than across tasks. On OOD-L2 the progress score awards one point for each of the three blocks placed and one for closing the drawer, out of four. Single cells carry roughly 10–15 points of noise at 10 rollouts, so we base conclusions on the per-condition averages, which aggregate 40 rollouts in distribution and 20 under L1, and on paired degradations. Because only pen→cup and carrot→bowl have an L1 variant, the degradation quoted in the main text compares each policy against its own score *on those two tasks* (80.0% for StellaVLA, 55.0% for StarVLA-OFT, 80.0% for $\pi_{0.5}$) rather than against the four-task in-distribution average. On OOD-L2 the progress scores are 1.9 for StellaVLA, 1.5 for $\pi_{0.5}$ and 1.1 for StarVLA-OFT; the task is absent from real-robot training, so only the human-hand take and its XR-retargeted counterpart in the pool hold a demonstration of it.

Baselines. The matched control is *StarVLA-OFT* (StarVLA Community, 2026), trained on the same real-robot episodes with the same backbone, action expert and action space, and differing only in receiving no demonstration and no auxiliary spatial-language expert. We additionally fine-tune the openpi flow-matching policy $\pi_{0.5}$ (Physical Intelligence et al., 2025) on the same episodes; as a separately pretrained model family it places our absolute numbers on an external scale rather than serving as a matched control.

C.1 Demonstration-Source Invariance

Because the reported rollouts already draw at random from all three sources, they cannot attribute an outcome to any one of them: each cell mixes sources across its ten placements. We therefore isolate the source offline on held-out teleoperated episodes that never enter training. Holding the observation fixed and swapping only the demonstration, we report the paired disagreement $D(S_1, S_2)$, the mean L_1 distance between the two predicted action chunks normalized by the per-dimension standard deviation σ of the ground-truth actions. Since the human and retargeted datasets are frame-aligned copies of one another, swapping between them varies the appearance of the embodiment while holding the trajectory, the timing and the task fixed.

The three sources are interchangeable under this test. Over all six pairs (Table 5; 1,780 in-distribution and 1,719 perturbed frames) any two disagree by 0.0014 to 0.0016σ, which is 0.02° to 0.03° of joint angle and under 0.05 mm of gripper width, with similar behaviour in and out of distribution. A bare human hand and the robot's own demonstration therefore steer the policy to nearly the same actions.

The comparison against no demonstration is weaker than this equivalence suggests. Withholding the demonstration moves the predicted action by only 0.0041–0.0045σ, about three times the source-to-source distance, with no frame exceeding 0.05σ. Demonstration dropout of 0.5 plausibly trains the action expert towards invariance to the context, and that expert reads the action tokens in parallel with the spatial-language expert, so the demonstration may act mainly through language. Since this measures single-step agreement rather than closed-loop success, and small consistent biases can compound over a rollout, it does not settle how much the demonstration contributes on hardware; the within-checkpoint rollout ablation that would settle it is left to future work.

Table 5: Paired disagreement D (in units of σ) between the action chunks a single fixed checkpoint predicts when only the demonstration source is swapped, on held-out teleoperated episodes (1,780 in-distribution and 1,719 perturbed frames). The three sources sit within 0.0016σ of one another, closer than any of them is to withholding the demonstration.

Demonstration pair	ID	OOD
D(real, none)	0.0041	0.0045
D(human, none)	0.0041	0.0044
D(xr, none)	0.0041	0.0044
D(human, real)	0.0015	0.0014
D(xr, real)	0.0015	0.0016
D(human, xr)	0.0014	0.0014

D Additional Ablations

Unless stated otherwise, ablations use the LIBERO checkpoint and report the four-suite average (AVG) and, where available, the LIBERO-Plus average (LP). "Eval-time surgery" denotes changing the demonstration fed to a *fixed* trained checkpoint at inference.

D.1 Demonstration Structure

Sub-goal granularity (Table 6) is nearly flat: three keyframes already reach 98.1 AVG, within 0.7 of the full ten (98.8). A plausible reading is that denser sampling adds frames but no new sub-goals, and it is the sub-goal decomposition the policy consumes, so returns saturate once the plan is complete rather than once the trajectory is densely covered. This is also what keeps the prefix short enough to cache.

Removing individual fields during training (Table 7) separates the two halves of the kinematic rationale. Dropping the 2D gripper path costs more (97.3) than dropping the 3D state (97.8), even though the 2D path is a projection of the same motion and carries no additional physical information; the projection is what ties the verbalized plan to the current image. The 0.5-point difference between the two is, however, comparable to rollout noise, so we do not rank them strongly, and the semantic sub-task label was not ablated separately.

Table 6: Sub-goal granularity. Best in **bold**.

Sub-goal frames	AVG
3	98.1
5	98.2
8	98.3
10 (default)	**98.8**

Table 7: Demonstration fields. Best in **bold**.

Fields	AVG
Full (default)	**98.8**
− 3D state	97.8
− 2D point	97.3

D.2 Auxiliary Language Loss

The language-loss weight λ (Table 8) trades the two objectives against each other. In distribution it follows an inverted U peaking at $\lambda = 0.3$ (98.8 AVG), while robustness falls monotonically with the weight, from 86.9 LP at $\lambda = 0$ to 81.9 at $\lambda = 1.0$.

One explanation fits both curves. A small language loss binds the shared representation to the sub-goal that motivates each displacement, which helps either objective. A large one makes that representation increasingly answerable to the annotation schema itself, to reproducing the phrasing our offline pipeline happened to generate, and the schema is a training artefact, so fitting it more tightly buys nothing under perturbation while consuming capacity. Worth noting is that the most robust setting is $\lambda = 0$: in-context conditioning alone already yields 86.9 LP, and the auxiliary loss is best understood as converting some of that robustness into in-distribution precision. We use $\lambda = 0.3$, which buys 1.2 AVG for 1.8 LP; because that choice was made on benchmark success rather than on a held-out validation split, the reported LIBERO figure is optimistically selected.

Table 8: Language-loss weight. Best per column in **bold**.

Weight λ	AVG	LP
0	97.6	**86.9**
0.3 (default)	**98.8**	85.1
0.6	97.5	84.0
1.0	97.2	81.9

D.3 Inference Efficiency

A single forward with no demonstration takes 64 ms on a single NVIDIA RTX 4090 (Table 9); an image+text demonstration costs 183 ms, which caching the immutable demonstration prefix reduces to 91 ms ($\sim 2\times$), and a text-only demonstration runs at 109 ms. Because the demonstration prefix is fixed within an episode, its cost is paid at the first step and not at any step after it, so a full multimodal demonstration ends up cheaper to carry than a text-only one without a cache. The price of in-context conditioning is thus close to a constant per episode rather than a tax per control step, although the cached prefix still adds 27 ms per step over the demonstration-free forward pass.

Decoding the language chain-of-thought is a different order of cost. In a single run on the same hardware, the action-only path runs at 88 ms per step, reproducing the cached image+text row; decoding the 83-token chain-of-thought before acting adds 3089 ms, leaving each step at 3177 ms. This is a factor of ~ 36, and roughly $6\times$ the 0.53 s that a 16-step chunk spans on the real robot at 30 Hz. Reasoning that must be emitted before acting is therefore not merely expensive but inadmissible at control rates. Our parallel experts avoid the choice: the spatial-language branch is left unexecuted at deployment at no cost to the action path, since the two experts read the same representation rather than one conditioning on the other.

Table 9: Inference latency. The last two rows were measured together in one run, in which the action-only path reproduces the 91 ms row above; decoding the chain-of-thought costs $\sim 36\times$ that path.

Configuration	KV cache	Latency (ms)
No demo	—	64
Image+Text	no	183
Image+Text	yes	91
Text-only	—	109
Language decoding, single run		
Action only	yes	88
+ CoT decode	yes	3177

E Discussion and Limitations

Language as a cross-embodiment bridge. A notable finding in our real-world experiments is the interchangeability of demonstration sources. When rendered into our structured, language-action payload, demonstrations from a bare human hand, an XR-retargeted arm, and the robot's own teleoperated episodes steer the policy to nearly identical actions (Table 5). The reported hardware success rates are themselves obtained under this regime, with each rollout's demonstration drawn at random from the merged three-source pool rather than pinned to the robot's own embodiment. This suggests that language can serve as a low-cost cross-embodiment interface: instead of training a latent action space or requiring paired human–robot datasets, the pretrained semantic grounding of large vision-language backbones can absorb embodiment gaps, which opens the possibility of using abundant, unstructured human video directly as in-context prompts for robotic control. The evidence is bounded, however, by the fact that the same checkpoint is only weakly sensitive to the demonstration in single-step prediction, so a source-controlled rollout study is needed to quantify the benefit on hardware.

Decoupling reasoning from latency. A major bottleneck in deploying reasoning-enriched policies, such as embodied chain-of-thought, is the autoregressive generation latency, which is incompatible with real-time control loops. By employing parallel experts and an asymmetric inference strategy, StellaVLA decouples the acquisition of physical reasoning from its execution. The backbone internalizes the structured rationale during training under joint language-action supervision, while at deployment the spatial-language expert is stripped away, so control incurs no autoregressive decoding overhead (Table 9). The ablation of λ shows that this supervision improves in-distribution success, whereas robustness to perturbations stems mainly from conditioning on structured demonstrations.

Limitations and future work. Despite its strong performance, our framework has several limitations. *Long-horizon compositionality:* while StellaVLA substantially improves task-level generalization and robustness, success rates on extremely long-horizon, multi-stage tasks (such as the Long Horizon suite in VLA-Arena) remain low for all evaluated models. The demonstration prefix is fixed for the duration of an episode, so it cannot re-plan once execution has drifted, and in-context conditioning alone therefore does not resolve multi-step error accumulation. *Dependence on retrieval quality:* the effectiveness of our in-context adaptation relies on retrieving a relevant expert demonstration. As the wrong-demonstration ablation shows, conditioning on a mismatched demonstration degrades performance below the no-demonstration baseline, so more robust retrieval under visual and semantic domain shift is a critical next step. Finally, *Coarse kinematic discretization:* our textualized kinematic rationale relies on a quantized representation of spatial movements to fit within the VLM's vocabulary. While this quantization suffices to shape the shared representation, it may introduce discretization errors that limit the precision of fine-grained manipulation. Continuous tokenization or hybrid representation schemes could mitigate this.