GENERALIZABLE VLA FINETUNING VIA REPRESENTATION ANCHORING AND LANGUAGE-ACTION ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Finetuning a pretrained vision-language model (VLM) on robot demonstrations via behavior cloning (BC) has become the standard recipe for vision-language-action (VLA) policies. However, BC finetuning progressively overwrites the pretrained representations that support visual and semantic generalization. Co-training on web image-text data, a common remedy, applies language and action losses to separate observations, leaving VLAs with language-action misalignment that standard manipulation benchmarks do not expose. We propose Anchor-Align, which augments BC with two objectives: *Vision-Language Anchoring* distills layer-wise representations from a frozen VLM copy to prevent this drift, while *Language-Action Alignment* converts each action target into a discrete motion-direction label and jointly trains language and action prediction on the same robot observation. We conduct real-world evaluations across eight manipulation settings on single-arm xArm7 and bimanual YAM robots, using two VLA architectures with regression and flow-matching action heads. Across these settings, Anchor-Align consistently improves over BC on novel targets, layouts, and motion-sensitive bimanual tasks requiring coordinated control. At scale in simulation, we demonstrate consistent improvements on OOD perturbations, perceptual robustness, and long-horizon control across LIBERO-PRO, LIBERO-Plus, and CALVIN, respectively, suggesting that preserving pretrained representations and effective action learning are not fundamentally at odds.

1 INTRODUCTION

Vision-Language-Action (VLA) models adapt pretrained vision-language models (VLMs) into robot manipulation policies by finetuning on expert demonstrations (Zitkovich et al., 2023; Kim et al., 2024). Action learning typically uses behavior cloning (BC), with either direct regression (Kim et al., 2025) or flow matching and diffusion (Black et al., 2025b). The premise is that such finetuning should transfer the VLM's semantic priors (understanding of spatial layout, directions, color, shape, etc.) to the resulting control policy. Realizing this promise requires more than retaining pretrained knowledge: finetuning must make it useful for control.

Consider a VLA policy finetuned to "pick up the green mug and place it on the plate" in a scene containing both a green and pink mug (Fig. 1). Since the two mugs share the same shape and manipulation affordances, the policy should generalize to "pick up the pink mug" if finetuning preserves the VLM's color understanding. However, with existing methods, it does not. In fact, BC finetuning corrupts the very prior that makes VLMs worth adapting, leading to two failure modes that persist even under good training practices.

First, standard BC optimizes only the action prediction loss, with nothing protecting the VLM's pretrained representations from being overwritten. Over the course of finetuning, these updates progressively erase the visuolinguistic and spatial concepts the VLM acquired during internet-scale pretraining. Concretely, on our physical xArm7 robot, a VLA trained to pick up the green mug reaches for the green mug 90% of the time even when instructed to pick up the pink one, indicating that color grounding has been erased from the backbone. On LIBERO-PRO's position-swap test, where object positions are rearranged at evaluation, the policy almost always replays its memorized training trajectory rather than acting on the current observation. To fix this failure mode,

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.

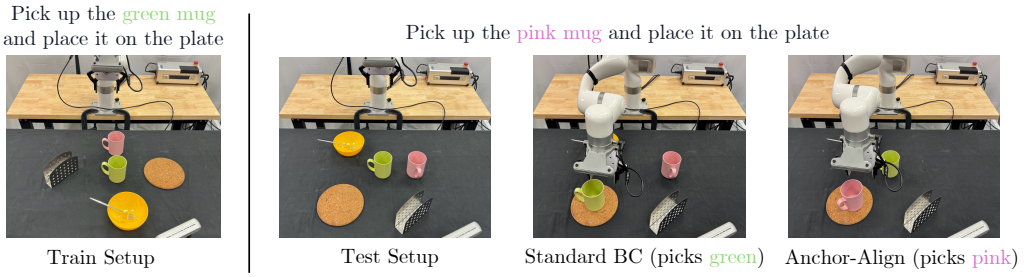


Figure 1: **Out-of-distribution test for VLAs.** Standard BC, finetuned to pick up the green mug, reaches for the green mug even when instructed to pick up the pink mug, whereas Anchor-Align retains the VLM's representations and successfully completes the task.

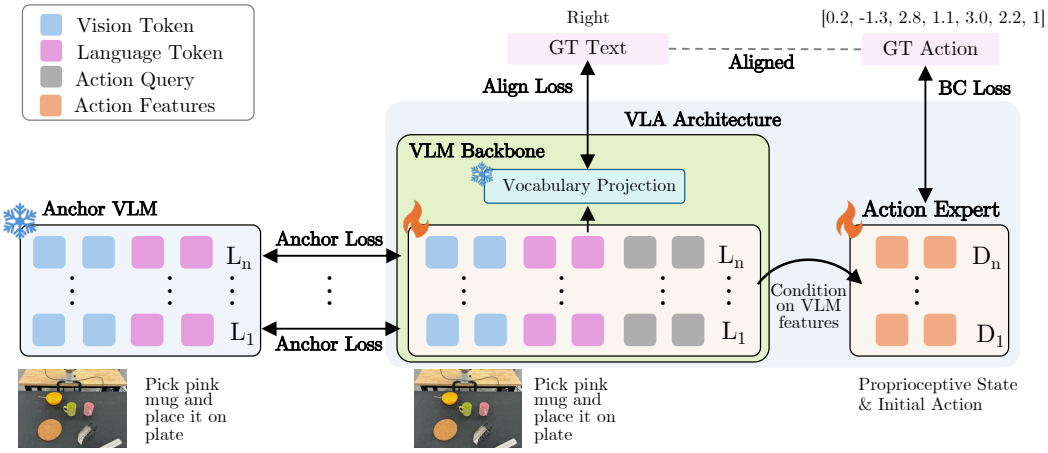


Figure 2: **Anchor-Align preserves VLM priors and aligns them with action during VLA finetuning.** Vision-Language Anchoring distills representations from a frozen pretrained VLM into the trainable VLA at every transformer layer (Anchor loss), preventing catastrophic forgetting. Language-Action Alignment programmatically converts each ground-truth demonstration trajectory into a discrete language label and trains the model to predict this label on the same observation it acts on (Align loss). Together, these objectives retain the pretrained representations while grounding action generation in VLM semantics.

recent work has proposed co-training, or augmenting continuous action data with general-purpose VQA, captioning, and scene-description data (Lee et al., 2025; Driess et al., 2025; Yang et al., 2025; Zhou et al., 2025c) to prevent catastrophic forgetting. Yet we find that on the position-swap test, co-trained VLAs score 0%. Evidently, losses on such auxiliary data do not transfer the VLM's semantic understanding to the finetuned VLA. As a more direct remedy, we propose *Vision-Language Anchoring*, or maintaining a frozen copy of the pretrained VLM (Fig. 2, left) and applying a distillation loss throughout finetuning. This requires no additional data or architectural changes, yet recovers out-of-distribution (OOD) generalization.

The second failure mode runs even deeper. Co-training gives the VLA's two heads disjoint supervision: the action head is supervised by expert demonstrations while the language head is supervised by losses on generic image-text tasks. Because the heads share a backbone but are never supervised on the same robot observation, their predictions may contradict each other: when the action head predicts a movement to the right, the language head may predict "left." To fix this misalignment and further improve task success, we introduce *Language-Action Alignment*, which co-supervises both heads on the same robot observation during finetuning (Fig. 2, top). To supply the aligned supervisory data, we automatically convert continuous action targets in expert demonstrations into discrete action directions expressed in language.

We propose combining Vision-Language Anchoring and Language-Action Alignment with the standard BC loss to form **Anchor-Align**, a VLA finetuning method with consistent improvements across simulation and real-world experiments under both regression and flow-matching action heads. In simulation, our method improves OOD generalization on LIBERO-PRO and LIBERO-Plus benchmarks, which introduce unseen spatial rearrangements, camera perturbations, and other shifts absent from training, as well as long-horizon control on CALVIN. In real-world experiments on an xArm7

robot, Anchor-Align improves spatial generalization, target generalization, and target grounding in clutter. On a bimanual YAM robot, it further improves motion-sensitive control, spanning reactive grasping and coordinated bimanual transport under visual shift.

2 Related Works

General-Purpose VLAs for Robot Manipulation. VLM-based policies leverage pretrained vision-language representations for instruction-conditioned control (Driess et al., 2023; Li et al., 2023; Qu et al., 2025b; Kim et al., 2025; Zitkovich et al., 2023). Recent general-purpose VLAs pair these pretrained backbones with expressive action-generation modules, including flow-matching action experts (Black et al., 2025b;a) and diffusion transformers (Bjorck et al., 2025). VLA-Adapter (Wang et al., 2025) enables direct adaptation without large-scale robotic pretraining through lightweight Bridge Attention, while StarVLA (StarVLA Community, 2026) provides a modular framework for combining backbones and action heads. We build on these two frameworks, finetuning pretrained VLMs directly on expert demonstrations.

Preserving VLM Representations in VLAs. Pretrained visual representations are a well-established driver of robot manipulation performance (Nair et al., 2022; Radosavovic et al., 2023; Dasari et al., 2023; Wang et al., 2024). In VLAs, this requirement extends beyond the vision encoder to the full VLM backbone, since semantic and spatial reasoning are distributed across both vision and language representations. Prior VLAs commonly mitigate catastrophic forgetting by co-training on robot and generic vision-language data (Yang et al., 2025; Lee et al., 2025; Qu et al., 2025a; Wen et al., 2025), with variants adding action-grounding objectives, perception tokens, or chain-of-thought plans (Yang et al., 2025; Lee et al., 2025; Zawalski et al., 2024; Zhou et al., 2025c;b). Grover et al. (2025) instead preserves pretrained features architecturally, anchoring a frozen vision encoder with a trainable one. MAPS (Huang et al., 2025) penalizes each module's weights for drifting from the pretrained checkpoint, constraining vision layers more tightly than action-oriented ones, and Kachaev et al. (2025) aligns visual patch features at a single intermediate layer with a frozen vision foundation model. Anchor-Align instead applies two simple losses at the VLM backbone level throughout finetuning, jointly preserving the pretrained VLM prior and enriching its representations with language supervision grounded in continuous robot actions. This couple's preservation with action-relevant adaptation, rather than treating preservation as an end goal.

Language supervision for action learning. Language supports action learning through intermediate predictions, action representations, and auxiliary supervision. RT-H (Belkhale et al., 2024) conditions low-level actions on predicted language motions, while ECoT (Zawalski et al., 2024) and InSpire (Zhang et al., 2025) introduce grounded reasoning chains and object–gripper spatial relations as textual intermediates. VLA-0 (Goyal et al., 2025) serializes discretized actions as text, whereas Actions-as-Language (Hancock et al., 2025) expresses low-level commands in natural language within a reasoning hierarchy. LAP (Zha et al., 2026) supervises structured action-chunk descriptions; its LAP-3B implementation pairs this objective with a continuous-action expert whose gradients are blocked from the VLM. Anchor-Align instead preserves pretrained representations while adapting the VLM backbone through continuous-action gradients. Vision-Language Anchoring constrains representational drift, while Language-Action Alignment grounds language predictions in the same action targets supervised by BC. Language thus supplements continuous-action learning rather than replacing it, with both objectives using only existing demonstrations and requiring no intermediate language generation at inference.

3 Anchor-Align Method

This section presents Anchor-Align (Fig. 2), a recipe that finetunes a pretrained VLM into a continuous action policy while preserving its pretrained VL representations. We first describe our base VLA architecture and training objective (Sec. 3.1), then the two objectives Anchor-Align adds on top of standard BC: *Vision-Language Anchoring* (Sec. 3.2) and *Language-Action Alignment* (Sec. 3.3).

3.1 Base VLA Architecture

We consider VLAs that adopt a pretrained vision-language backbone with a dedicated action head (StarVLA Community, 2026; Wang et al., 2025), as shown in Figure 2. The backbone encodes the visual observation and language instruction into a hierarchy of multimodal representations. Separately, robot proprioception is encoded by a lightweight state projector and provided to the action

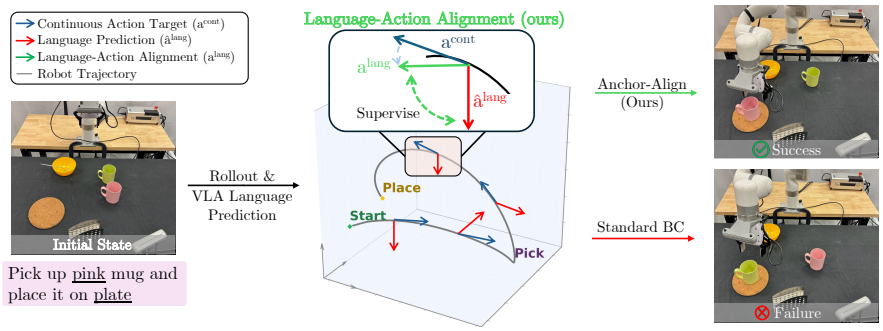


Figure 3: **Anchor-Align VLA aligns the VLM's language predictions with action generation during VLA finetuning.** Standard BC supervises only action prediction, allowing the model's language output (red) to contradict the required motion on the same observation (blue). Language-Action Alignment derives a discrete motion-direction label from each ground-truth action target and trains the backbone to predict it on the same observation (green), bringing language and action into agreement.

pathway as a conditioning signal. A fixed set of learned action queries extracts task-relevant information from the vision-language representations, either through joint self-attention within the backbone or through cross-attention in the action head. The action head then combines the resulting query features with the encoded robot state to predict a fixed-horizon chunk of continuous 7-DoF actions (3D translation and rotation offsets plus gripper control) using either direct regression or conditional flow matching.

We train this architecture with three objectives. The primary term is the standard BC loss $\mathcal{L}_{\text{action}}$ between the predicted and ground-truth action chunks, $\hat{\mathbf{a}}^{\text{cont}}$ and $\mathbf{a}^{\text{cont}}$ respectively. The form of the BC loss depends on the action-head architecture: an L_1 loss for a regression action head, and an L_2 loss over the predicted velocity for a flow-matching action head. To it we add two objectives, both evaluated on the same robot observation: an anchoring loss $\mathcal{L}_{\text{anchor}}$ that preserves the backbone's pretrained vision-language representations (Sec. 3.2), and an alignment loss $\mathcal{L}_{\text{align}}$ that grounds the model's language predictions in its actions (Sec. 3.3). The three terms are optimized jointly:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \lambda_{\text{anchor}}\,\mathcal{L}_{\text{anchor}} + \lambda_{\text{align}}\,\mathcal{L}_{\text{align}}, \tag{1}$$

where λ_{anchor} and λ_{align} weight the two added objectives against the action loss.

3.2 Vision-Language Anchoring

To prevent catastrophic forgetting, we anchor the VLA's backbone VLM with a frozen copy of the same VLM (the *anchor* VLM), which processes the same input batch (images, text) in parallel. This distills the original pretrained VLM's representations into the trainable backbone. In effect, the anchor VLM serves as a frozen teacher, a role well established in the ML literature: feature distillation trains a student to match a teacher's representations (Hinton et al., 2015; Romero et al., 2015; Jiao et al., 2020), and anti-forgetting methods regularize a finetuned model toward its pretrained copy in weight space (Kirkpatrick et al., 2017; Li et al., 2018) or representation space (Li & Hoiem, 2018; Douillard et al., 2020; Mukhoti et al., 2024). We anchor hidden states of the vision and text tokens between the backbone and anchor VLM at every decoder layer $u \in \mathcal{D}$, where $\mathcal{D}$ is the set of decoder layers. Let $\mathbf{m}$ denote the set of vision and text positions. The anchoring loss is the mean squared error (MSE) between the two sets of hidden states, averaged over all $|\mathcal{D}|$ decoder layers:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{|\mathcal{D}|}\sum_{u\in\mathcal{D}}\frac{1}{Nd}\sum_{i\in\mathbf{m}}\sum_{j=1}^{d}\left(\mathbf{H}_u^S[i,j] - \mathbf{H}_u^A[i,j]\right)^2, \tag{2}$$

where $\mathbf{H}_u^S$ and $\mathbf{H}_u^A$ are the hidden states of the trainable backbone (S) and the anchor (A) at decoder layer u, $\mathbf{H}_u[i,j]$ is the j-th feature of the token at position i, $N = |\mathbf{m}|$ is the number of vision and text tokens, and d is the hidden dimension of the VLM.

3.3 Language-Action Alignment

As illustrated in Fig. 3, we programmatically convert the continuous action target $\mathbf{a}^{\text{cont}}$ into a motion-direction label $\mathbf{a}^{\text{lang}}$ for each non-stationary demonstration chunk and use it to supervise the VLM backbone on the same observation used for continuous action prediction. We use the last-layer

hidden state of the last instruction token, immediately before action prediction, which summarizes the visual prefix and language instruction. With N vision and text tokens, this pre-action hidden state is $\mathbf{h}^{\text{pre}} = \mathbf{H}^S[N] \in \mathbb{R}^d$, where $\mathbf{H}^S$ denotes the backbone VLM's last-layer hidden states. We project $\mathbf{h}^{\text{pre}}$ onto the vocabulary through a learned projection and the frozen pretrained language head: $\hat{\mathbf{a}}^{\text{lang}} = \mathbf{W}_{\text{lm}}\mathbf{W}_{\text{proj}}\mathbf{h}^{\text{pre}} \in \mathbb{R}^{|\mathcal{V}|}$, where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d\times d}$ is learned and $\mathbf{W}_{\text{lm}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the frozen pretrained LM head. The alignment loss is the cross-entropy between these vocabulary logits and the ground-truth direction label:

$$\mathcal{L}_{\text{align}} = \text{CE}(\hat{\mathbf{a}}^{\text{lang}}, \mathbf{a}^{\text{lang}}). \tag{3}$$

Language Label Construction. The alignment targets are derived from ground-truth action trajectories with no human annotation. For each training sample, we assign one of six interpretable direction labels, $W = \{\textit{up}, \textit{down}, \textit{left}, \textit{right}, \textit{forward}, \textit{backward}\}$, through three steps: average chunking, filtering, and discretization. Given the batch of action targets $\mathbf{A} \in \mathbb{R}^{B\times K\times 7}$ ($\mathbf{a}^{\text{cont}}$ stacked over B batch samples, each a K-step chunk), we average the translational components across the chunk dimension to reduce action noise: $\bar{\mathbf{v}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{A}_{:,k,1:3} \in \mathbb{R}^{B\times 3}$. We then remove near-stationary samples satisfying $\|\bar{\mathbf{v}}_i\|_2 < \tau$ (where i indexes samples in the batch), with τ a small stationarity threshold. For each remaining sample, we choose the dominant translation axis, $j^\star = \arg\max_{j\in\{x,y,z\}} |\bar{v}_{i,j}|$, where $\bar{v}_{i,j}$ is the j-th scalar component of $\bar{\mathbf{v}}_i$, and use the sign of $\bar{v}_{i,j^\star}$ to select the corresponding direction word. The resulting direction label $\mathbf{a}^{\text{lang}}$ supervises the language-alignment loss. Training implementation details are given in App. B.1; the diagnostic evaluation using these direction labels is described in App. E.1.

4 Experiments

We evaluate on simulation benchmarks and real-world robot experiments. Sec. 4.1 describes the benchmarks, baselines, and implementation details. Sec. 4.2 and Sec. 4.3 report results and ablations on LIBERO-PRO, LIBERO-Plus, and CALVIN. Sec. 4.4 results on an xArm7 and a bimanual YAM.

4.1 Experimental Setup

Benchmarks. We evaluate on LIBERO (Liu et al., 2023), LIBERO-PRO (Zhou et al., 2025a), LIBERO-Plus (Fei et al., 2025), and CALVIN. LIBERO comprises four 10-task suites (Spatial, Object, Goal, Long) whose test setups follow the training distribution, probing only in-distribution performance and now being largely saturated. We therefore report standard LIBERO results in App. C.2, where Anchor-Align VLA outperforms all baselines, and focus below on the more challenging LIBERO-PRO and LIBERO-Plus stress tests. LIBERO-PRO (Zhou et al., 2025a) evaluates *semantic generalization* under OOD perturbations: language rephrase, object swap, and position swap. LIBERO-Plus (Fei et al., 2025) evaluates *perceptual robustness* across seven axes reflecting real-world deployment variability: camera viewpoint, lighting, background texture, object layout, robot initial state, language instruction, and sensor noise. Subcategory details are provided in App. C.1. Finally, CALVIN ABC→D tests long-horizon, language-conditioned manipulation by training on environments A, Bx, and C and evaluating zero-shot on held-out environment D. We further extend our study by collecting a dataset of more than 300 real-world demonstrations. We use this dataset to evaluate performance across 8 manipulation tasks.

Baselines. We compare five baseline families (details in App. A): (i) the VLA models OpenVLA-OFT (Kim et al., 2025) and VLA-Adapter (Wang et al., 2025) (our base model); (ii) the representation-preserving baselines VLA-Adapter[Frozen] (Wang et al., 2025), which freezes the backbone, and MAPS (Huang et al., 2025), our reimplementation of module-wise proximity scheduling on VLA-Adapter; (iii) the co-trained VLAs: MolmoAct (Lee et al., 2025) and *Co-training + KI*, our reimplementation of Knowledge Insulation (Driess et al., 2025) on VLA-Adapter; (iv) the language-based action baselines VLA-0 (Goyal et al., 2025), which represents discretized actions as text, and LAP (Zha et al., 2026), which supervises language descriptions of action-chunk motion; and (v) the CALVIN baselines UniVLA (Bu et al., 2025) and OpenHelix (Cui et al., 2025).

Implementation Details. We use Prismatic-Qwen2.5-0.5B (Yang et al., 2024) as VLM backbone and finetune it with LoRA (Hu et al., 2022) with rank $r = 64$, applied on all layers. The input sequence is vision + text + proprioceptive state. We use a regression action head with bridge attention to layer-wise condition the action head latents on the VLM features (Wang et al., 2025). Image encoding, hyperparameters, and other implementation details are provided in App. B. We additionally evaluate StarVLA (StarVLA Community, 2026), which pairs Qwen2.5-VL (Bai et al., 2025) with a flow-matching GR00T FM-DiT (Bjorck et al., 2025) action head, in our real-world experiments.

Table 1: **Success rates on the LIBERO-PRO and LIBERO-Plus benchmarks.** Anchor-Align VLA achieves the highest mean on both benchmarks. The best result in each column is **bolded** and the second-best is underlined. The 5-seed mean and variance are reported in App. C.4, confirming that these gains are statistically significant.

Method	LIBERO-PRO				LIBERO-Plus							
	Lang. Reph.	Object Swap	Pos. Swap	Mean	Lang. Instr.	Bg. Text.	Robot Init	Cam. View	Obj. Layout	Light Cond.	Sensor Noise	Mean
MolmoAct (Lee et al., 2025)	77.8	82.4	0.0	53.4	79.5	84.1	47.4	10.1	76.5	77.4	53.4	59.9
VLA-0 (Goyal et al., 2025)	75.4	93.0	5.4	57.9	60.0	70.2	34.9	42.6	83.9	88.7	60.1	62.0
OpenVLA-OFT (Kim et al., 2025)	74.4	95.2	0.0	56.5	81.5	95.7	40.3	94.7	88.6	95.5	28.2	74.1
VLA-Adapter (Wang et al., 2025)	91.1	89.6	2.3	61.0	85.1	90.7	52.6	92.6	93.2	93.2	89.5	85.1
VLA-A [Frozen] (Wang et al., 2025)	56.0	73.4	0.0	43.1	41.5	70.9	35.1	94.4	62.3	84.9	36.2	59.9
Co-training + KI (Driess et al., 2025)	53.6	92.0	0.0	48.5	61.5	93.8	48.6	89.6	77.1	96.9	48.4	72.4
LAP (Zha et al., 2026)	93.2	95.0	12.4	66.9	**88.1**	99.2	54.0	72.6	92.5	95.5	76.1	81.8
MAPS (Huang et al., 2025)	93.0	90.2	11.6	64.9	87.1	97.3	52.9	95.2	87.5	97.3	90.6	86.3
Anchor-Align VLA	**97.0**	**96.2**	**22.6**	**71.9**	87.2	**99.6**	**59.1**	**96.3**	**97.4**	**99.0**	**96.9**	**90.3**

4.2 Simulation Benchmark Comparison

Anchor-Align improves visual grounding and spatial understanding.

Tab. 1 shows that Anchor-Align VLA achieves the highest mean on both LIBERO-PRO and LIBERO-Plus. For LIBERO-PRO, the position-swap axis is the hardest: task-relevant objects are rearranged, so a policy that memorized fixed scene-to-action mappings fails. Most baselines fail almost entirely under this setting, while even the strongest alternatives reach only 12.4%; in contrast, Anchor-Align VLA achieves 22.6%. Fig. 4 shows qualitative examples from LIBERO-PRO, where Anchor-Align VLA generalizes to the perturbed scene, while standard BC replays its training trajectory.

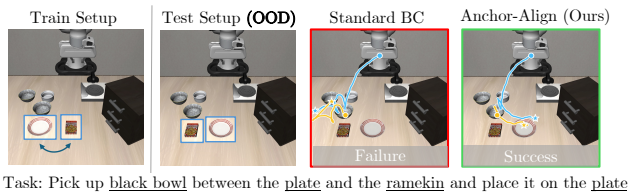


Figure 4: LIBERO-PRO Swap, performance of Anchor-Align in OOD condition.

Anchor-Align improves robustness to perturbations. LIBERO-Plus consists of challenging perturbations like changing background texture, adding sensor noise, or unseen initial robot arm configurations. As we can see from Tab. 1, Anchor-Align VLA improves over VLA-Adapter (Wang et al., 2025) on every axis, with the largest gains on the hardest perturbations: background texture ($+8.9$), sensor noise ($+7.4$), robot initial state ($+6.5$), and lighting ($+5.8$). In absolute terms, Anchor-Align VLA succeeds 99.6% under background texture changes and 99.0% under lighting changes. Per-suite robustness breakdowns for the LIBERO Long, Object, and Goal suites in App. C.3.

Table 2: Long-horizon generalization on CALVIN ABC→D

CALVIN ABC→D	1/5 (↑)	2/5 (↑)	3/5 (↑)	4/5 (↑)	5/5 (↑)	Len (↑)
UniVLA (Bu et al., 2025)	95.5	85.8	75.4	66.9	56.5	3.8
OpenVLA-OFT (Kim et al., 2025)	96.3	89.1	82.4	75.8	66.5	4.1
OpenHelix (Cui et al., 2025)	97.1	91.4	82.8	72.6	64.1	4.1
VLA-Adapter (Wang et al., 2025)	98.3	94.0	87.5	80.0	73.1	4.3
Anchor-Align VLA	**99.1**	**95.8**	**90.6**	**84.7**	**77.9**	**4.5**

Anchor-Align improves long-horizon generalization. Tab. 2 shows results on the long-horizon CALVIN ABC→D benchmark. In each CALVIN rollout, the policy receives a chain of five language instructions and must complete them consecutively. The $k/5$ columns report the fraction of rollouts that complete the first k instructions, and **Len** denotes the average number of tasks completed consecutively (out of five). Because the chain advances only if each instruction succeeds, small grounding errors compound over the rollout. We therefore compare against the baselines most relevant to this claim: OpenVLA-OFT and UniVLA, large-scale action-pretrained VLAs; OpenHelix, a recent CALVIN-specific VLA; and VLA-Adapter trained with standard BC. Anchor-Align VLA is the best at every chain length and attains the longest average rollout, with the gain over VLA-Adapter widening on the deepest chains where errors compound, from $+0.8$ at one instruction to $+4.8$ at five. The results indicate preserving and aligning pretrained VLM representations improves long-horizon generalization.

4.3 Impact of Anchoring and Alignment

Both anchoring and alignment are necessary. Tab. 3 presents an ablation analysis that isolates the impact of our proposed additional loss terms by comparing the VLA-Adapter baseline with Anchor VLA (only the layer-wise MSE distillation loss $\mathcal{L}_{\text{anchor}}$), Align VLA (only the language-action alignment loss $\mathcal{L}_{\text{align}}$), and the full Anchor-Align VLA. We can see that each of anchoring and alignment independently improve over standard BC on every axis, and combining them in Anchor-Align VLA exceeds either alone. Thus, the two play complementary roles: anchoring preserves the representational substrate that alignment leverages.

Table 3: Ablations and semantic-label controls on LIBERO-PRO and LIBERO-Plus.

Method	Key Technique	PRO	Plus
VLA-Adapter	*Standard BC*	61.0	85.1
Align VLA	*Alignment only*	65.9	88.6
Anchor VLA	*Anchoring only*	68.1	87.3
Shuffle	*Control*	61.4	84.9
Scatter	*Control*	63.3	85.7
Anchor-Align VLA	*Full method*	**71.9**	**90.3**

Semantic alignment beyond auxiliary classification. We test whether the gains from Language-Action Alignment depend on the semantic correspondence between demonstrated motion and its language label, rather than simply supervising direction prediction. We retain the same training setup, including anchoring and the frozen language head, but change the target words assigned to the six motion directions. *Shuffle* applies a fixed permutation of the six direction words (e.g., forward $\rightarrow$ right), preserving directional vocabulary while breaking its correspondence to the demonstrated motion. *Scatter* assigns each direction a fixed word unrelated to motion (e.g., forward $\rightarrow$ purple). Both controls retain a one-to-one mapping between motion directions and target words, requiring the model to distinguish the same directions without semantically matching words to actions. As shown in Tab. 3, both perform substantially worse than the semantically matched formulation, indicating that direction prediction alone is insufficient and that the gains depend on the correct correspondence between motion and language (details in App. B.5). Linear probes further show improved motion-direction decoding from the backbone (App. D.5). Rollout diagnostics show higher language–action agreement for Anchor-Align, positively correlated with task success (App. E.2).

Anchor-Align preserves VLM representations. For effective generalization, a VLA must be able to perform visual reasoning, i.e., understand objects, their semantics, and their attributes in a scene, an ability that is a hallmark of pretrained VLM representations. GQA (Hudson & Manning, 2019) tests this ability through compositional questions spanning object, attribute, and spatial understanding. Here, we quantify visual reasoning in standard BC and Anchor-Align, through GQA accuracy over the course of finetuning with the frozen VLM (same VLM used as backbone for VLA model) as an upper bound. As shown in Fig. 5, standard BC degrades monotonically, losing 94% of its GQA accuracy within 10K steps (catastrophic forgetting), whereas Anchor-Align retains roughly 65% of the pretrained accuracy. Correspondingly, LIBERO-PRO task success rates (Table 1) are 71.9% (Anchor-Align), 61.0% (standard BC), and 43.1% (frozen), showing that action performance and vision-language reasoning are not fundamentally at odds. We further substantiate this study by quantifying the change in representation by different finetuning paradigms in App. D.3.

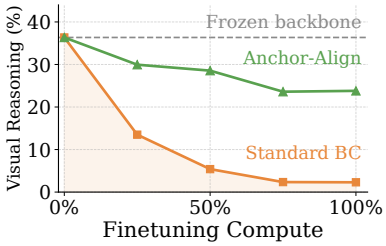


Figure 5: Visual reasoning accuracy during finetuning. The frozen VLM is the upper bound (dashed).

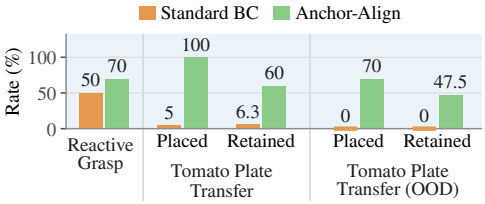
4.4 Real-World Results

We extend our study to real-world robotic manipulation through controlled experiments across two distinct VLA architectures. We compare Anchor-Align against standard BC using VLA-Adapter (Wang et al., 2025), which pairs a Prismatic-Qwen2.5-0.5B backbone with a Bridge-Attention regression head, and StarVLA (StarVLA Community, 2026), which pairs a Qwen2.5-VL backbone with a GR00T FM-DiT flow-matching action head. These models span different VLM scales and action-generation architectures. We evaluate four capabilities: (1) target generalization, (2) spatial generalization, (3) target grounding in clutter, and (4) motion understanding and stability.

We evaluate these capabilities across two robotic embodiments. The first three capabilities are evaluated on a single-arm 7-DoF UFactory xArm7, while motion understanding and stability are evaluated on the 14-DoF bimanual YAM platform, where the tasks require reactive execution and coordinated motion across both arms. For each evaluated condition, we run 20 rollouts per method and report re-



(a) `Reactive Grasp`. (b) `Tomato Plate Transfer`. (c) `Tomato Plate Transfer (OOD)`.



(d) Standard BC versus Anchor-Align on VLA-Adapter.

Figure 6: **Motion Understanding and Stability on YAM. (a–c)** Task setups; **(d)** Bars show success rate for `Reactive Grasp`, plate-placement success (*Placed*), and mean percentage of tomatoes retained (*Retained*).

sults as percentages. Each comparison uses the same underlying VLA architecture for Anchor-Align and standard BC, differing only in the training objective. Hardware, data collection, and deployment details are provided in App. B.4, with limitations and error analysis in App. F.

Motion Understanding and Stability. In Anchor-Align, the *Language-Action Alignment* loss is applied independently to each arm, with a separate motion-direction label for each. We compare Anchor-Align against Standard BC across three tasks. 1) `Reactive Grasp` requires the robot to intercept and grasp a bottle carried by a moving person, adapting its reach and gripper closure to the target motion. The bottle can vary in speed, approach direction, height, and holding pose (Feng et al., 2026), requiring the policy to estimate and anticipate its motion rather than react only to its instantaneous location. 2) `Tomato Plate Transfer` requires the robot to lift and transport a plate carrying eight tomatoes onto a box while maintaining a level bimanual grasp and smooth motion. Height mismatch between the arms can tilt the plate, while abrupt motion can cause the tomatoes to roll off. As noted by (Song et al., 2026), this task is challenging even for human teleoperators because these constraints must be maintained jointly throughout the transfer. 3) `Tomato Plate Transfer (OOD)` evaluates the same task under an unseen visual configuration, with the box covered in white paper and the surrounding workspace walls removed.

With VLA-Adapter on YAM, Anchor-Align VLA improves `Reactive Grasp` success from 50% to 70%. On `Tomato Plate Transfer`, plate-placement success increases from 5% to 100%, while mean tomato retention increases from 6.3% to 60%. Under the unseen visual configuration, placement success increases from 0% to 70% and mean tomato retention from 0% to 47.5% (Fig. 6d). These gains indicate substantially improved motion-sensitive control, spanning both reactive grasping of moving targets and coordinated bimanual transport. We further quantify the latter in Appendix G.1, where Anchor-Align reduces the mean inter-arm height-change mismatch by 46.2%.

Target Generalization (OOD). We evaluate target generalization on the xArm7 across two tasks. 1) `Pink Mug Placement` tests an OOD change in the language-specified object together with an unseen spatial layout. During training, the policy receives the instruction *pick up the green mug and place it on the plate*; at test time, the instructed target changes to the pink mug and the objects are presented in a spatial configuration not observed during training (Fig. 8a). Successful execution therefore requires grounding the test-time language to the specified color and object while localizing that target under an unseen layout `Cable Insertion` tests an OOD change in the language-specified receptacle. Across five training configurations that contain a red cable manager, the policy is instructed to insert the wire into it. At test time, the red cable manager is replaced by a green one and the policy is instructed to insert the wire into the green cable manager. This additionally requires precise target grounding because successful insertion demands high spatial precision.

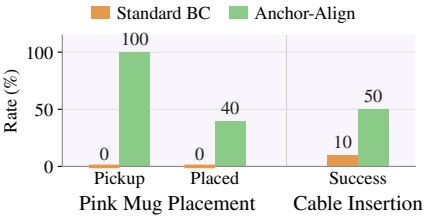
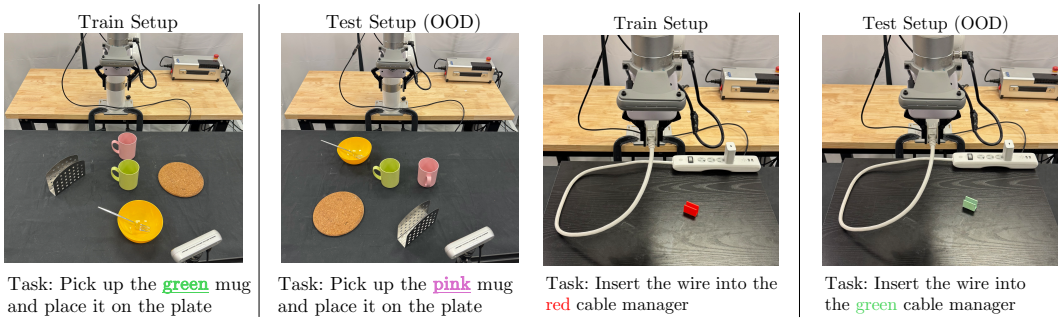


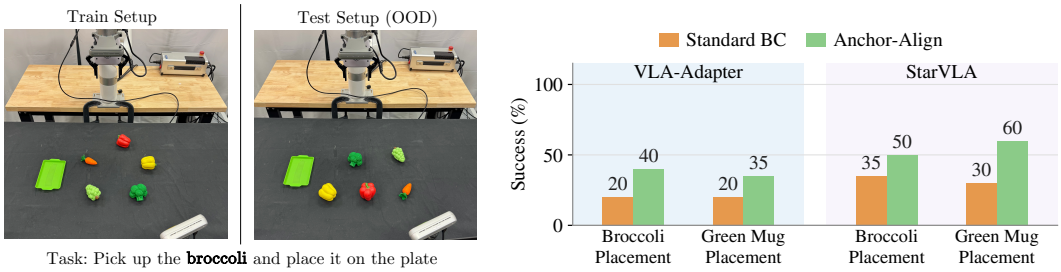
Figure 7: Results with StarVLA architecture.

With StarVLA, standard BC continues to act on the training target in `Pink Mug Placement`: it picks up the green mug in 90% of trials and places it on the plate in 30%. In contrast, Anchor-Align VLA picks up the instructed pink mug in 100% of trials and places it in 40%. This distinguishes successful transfer to the instructed target from completing the subsequent placement, which remains more challenging. On `Cable Insertion`, Anchor-Align VLA achieves 50% success at inserting the wire into the instructed green cable manager, compared with 10% for standard BC (Fig. 7).



(a) `Pink Mug Placement`. (b) `Cable Insertion`

Figure 8: **Target Generalization (OOD) on xArm7. (a)** Target-object and spatial-layout changes between train & test. **(b)** Target-receptacle color shift changes between train & test.



(a) `Broccoli Placement`. (b) Success rates.

Figure 9: **Spatial Generalization (OOD). (a)** `Broccoli Placement` training and test setups. **(b)** Success rates for `Broccoli Placement` and `Green Mug Placement` across both architectures.

Spatial Generalization (OOD). We evaluate spatial generalization to unseen spatial configurations on two backbones: VLA-Adapter and StarVLA. `Broccoli Placement` changes the position and orientation of the broccoli and the surrounding distractor objects relative to the training configurationsFig. 9a. `Green Mug Placement` introduces larger workspace rearrangements, relocating the target and surrounding objects, including distractors, to configurations unseen during training. We reuse the train/test spatial configurations shown in Fig. 8a; unlike `Pink Mug Placement`, however, the instruction and target identity remain fixed: *pick up the green mug and place it on the plate*. On `Broccoli Placement`, Anchor-Align VLA increases success from 20% to 40% with VLA-Adapter and from 35% to 50% with StarVLA. On `Green Mug Placement`, success increases from 20% to 35% with VLA-Adapter and from 30% to 60% with StarVLA (Fig. 9). These improvements across both architectures demonstrate better generalization to unseen spatial configurations while retaining the same instruction and manipulation objective.

Target Grounding in Clutter. We evaluate target grounding in clutter with `Clutter Scene` using both VLA-Adapter and StarVLA. The robot must pick up the object specified by the language instruction from among multiple distractors. We vary the target across three objects: `pineapple`, `yellow pepper`, and `red pepper`, requiring the policy to ground different instructions within the same scene. On `Clutter Scene`, Anchor-Align VLA increases success from 40% to 85% with VLA-Adapter and from 45% to 70% with StarVLA (Fig. 10). These gains show that Anchor-Align VLA more reliably identifies and manipulates the language-specified target despite surrounding distractors.

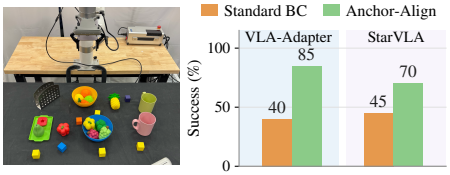


Figure 10: Success rates for `Clutter Scene`.

5 Conclusion

Standard BC can overwrite pretrained VLM semantics, while co-training does not ensure language-action alignment. Anchor-Align addresses both through representation anchoring and language-action alignment losses, consistently improving OOD generalization, robustness, and real-world manipulation across VLA architectures and action heads without additional data or annotation. These results show that preserving and grounding VLM priors improves generalization and action learning.

REPRODUCIBILITY STATEMENT

We will release a complete reference implementation of Anchor-Align under an open-source license, including: (1) implementations of Vision-Language Anchoring and Language-Action Alignment for the evaluated VLA architectures; (2) training and evaluation code with exact configurations for LIBERO, LIBERO-PRO, LIBERO-Plus, and CALVIN; (3) scripts for the representation, language-action alignment, and real-world analyses reported in the paper; and (4) the real-world demonstration datasets collected for all evaluated tasks. We will also release the trained model checkpoints used for the reported experiments. Detailed architectural specifications, hyperparameters, training procedures, compute requirements, and real-world implementation details are provided in Appendix B, with additional evaluation protocols and statistical analyses in Appendices C–E and G.

AI USE STATEMENT

We used generative AI tools solely to assist with grammar correction and to improve the clarity, readability, and presentation of author-written text. The tools were not used to generate or modify research ideas, methodology, experiments, results, or scientific claims, and were instructed to preserve technical terminology, method names, notation, and other domain-specific content. All AI-assisted edits were reviewed and approved by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quan Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. RT-H: Action Hierarchies Using Language. *arXiv preprint arXiv:2403.01823*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi 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.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL `https://proceedings.mlr.press/v305/black25a.html`.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems*, 2025b.

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.

Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, Han Zhao, Siteng Huang, and Donglin Wang. OpenHelix: A Short Survey, Empirical Analysis, and Open-Source Dual-System VLA Model for Robotic Manipulation. *arXiv preprint arXiv:2505.03912*, 2025.

Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An Unbiased Look at Datasets for Visuo-Motor Pre-Training. In *Conference on Robot Learning*, pp. 1183–1198, 2023.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *European Conference on Computer Vision*, 2020.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An Embodied Multimodal Language Model. *arXiv preprint arXiv:2303.03378*, 2023.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 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.

Zhenyang Feng, Jimin Heo, Erik B. Sudderth, and Unnat Jain. TEMPO: Learning Temporal Context for Dynamic Robot Manipulation. *arXiv preprint arXiv:2609.16864*, 2026.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. VLA-0: Building State-of-the-Art VLAs with Zero Modification. *arXiv preprint arXiv:2510.13054*, 2025.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I. Christensen, Hao Su, and Xuanlin Li. Enhancing Generalization in Vision-Language-Action Models by Preserving Pretrained Representations. *arXiv preprint arXiv:2509.11417*, 2025.

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. URL `https://arxiv.org/abs/2509.22195`.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

Chengyue Huang, Mellon M. Zhang, Robert Azarcon, Glen Chou, and Zsolt Kira. MAPS: Preserving Vision-Language Representations via Module-Wise Proximity Scheduling for Better Vision-Language-Action Generalization. *arXiv preprint arXiv:2511.19878*, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K Kovalev, and Aleksandr I Panov. Don't Blind Your VLA: Aligning Visual Representations for OOD Generalization. *arXiv preprint arXiv:2510.25616*, 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. In *Conference on Robot Learning*, 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.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

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.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit Inductive Bias for Transfer Learning with Convolutional Networks. In *International Conference on Machine Learning*, 2018.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

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

Jishnu Mukhoti, Yarin Gal, Philip H. S. Torr, and Puneet K. Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *Transactions on Machine Learning Research*, 2024.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, 2022.

Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Dong Wang, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, Maoqing Yao, Haoran Yang, Jiacheng Bao, Bin Zhao, and Xuelong Li. EO-1: An Open Unified Embodied Foundation Model for General Robot Control. *arXiv preprint arXiv:2508.21112*, 2025a.

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. *arXiv preprint arXiv:2501.15830*, 2025b.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-World Robot Learning with Masked Visual Pre-training. In *Conference on Robot Learning*, pp. 416–426, 2023.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations*, 2015.

Yuchen Song, Aditya Mittal, and Unnat Jain. Learning beyond what humans can demonstrate. *arXiv preprint arXiv:2609.24996*, 2026.

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

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling Proprioceptive-Visual Learning with Heterogeneous Pre-trained Transformers. In *Advances in Neural Information Processing Systems*, 2024.

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*, 2025.

Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. DexVLA: Vision-Language Model with Plug-In Diffusion Expert for General Robot Control. *arXiv preprint arXiv:2502.05855*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A Foundation Model for Multimodal AI Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14203–14214, 2025.

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. URL `https://arxiv.org/abs/2602.10556`.

Ji Zhang, Shihan Wu, Xu Luo, Hao Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. InSpire: Vision-Language-Action Models with Intrinsic Spatial Reasoning. *arXiv preprint arXiv:2505.13888*, 2025. URL `https://arxiv.org/abs/2505.13888`.

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*, 2025a.

Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. ChatVLA-2: Vision-Language-Action Model with Open-World Embodied Reasoning from Pretrained Knowledge. *arXiv preprint arXiv:2505.21906*, 2025b.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and Feifei Feng. ChatVLA: Unified Multimodal Understanding and Robot Control with Vision-Language-Action Model. *arXiv preprint arXiv:2502.14420*, 2025c.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen,

Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.