VLA-Trace: Diagnosing Vision-Language-Action Models through Representation and Behavior Tracing

Anonymous ACL submission

Abstract

Understanding how Vision-Language-Action (VLA) models transform multimodal knowledge into embodied control remains an open challenge. We present VLA-Trace, a progressive diagnostic framework that analyzes VLA models through a unified evidence chain from representation dynamics to causal control attribution and behavioral manifestation. It specifically combines cross-modal and checkpoint-drift centered kernel alignment (CKA) to trace representation evolution, attention knockout interventions to identify modality-specific control pathways, and rollout-level behavioral probes to examine grounding, shortcut dependence, and semantic following. Experiments on $\pi_{0.5}$ and OpenVLA reveal three key findings. First, the two models exhibit distinct modality-specific adaptation dynamics during VLA finetuning. Second, they rely on different multimodal routing strategies and layer-wise dependencies during action decoding. Third, although VLA policies excel at visually grounded trajectory generation, they remain limited in fine-grained semantic following. These findings highlight future directions for representation-preserving adaptation, causal VLA circuits, and compositional semantic control.

1 Introduction

Vision-Language-Action (VLA) models have become a promising paradigm for advancing embodied intelligence in real-world environments. Built upon large-scale pretrained vision-language models (VLMs) (Beyer et al., 2024; Touvron et al., 2023; Xiao et al., 2023), recent systems, including X-VLA (Zheng et al., 2025), OpenVLA-style models (Kim et al., 2024, 2025), and π-style models (Black et al., 2026; Pertsch et al., 2025; Intelligence et al., 2025b,a), achieve strong robotic control performance by unifying visual-language perception and action generation within a single policy framework. However, despite these successes, the internal mechanisms underlying action generation remain poorly understood. This issue is critical because, without understanding how robotic policy learning reshapes internal representations, it is difficult to diagnose model failures and, consequently, to design more effective VLA architectures. In particular, it remains unclear whether multimodal knowledge is preserved, how visual and linguistic signals are aligned during policy learning, and which modalities govern action decoding.

To answer these questions, we propose **VLA-Trace**, a systematic three-stage analysis framework that progressively investigates VLA models from latent representation evolution to causal control attribution and finally to behavioral dependence. As illustrated in Fig. 1, instead of treating the policy as an end-to-end system, we construct a unified evidence chain for interpreting VLA adaptation. First, we employ cross-modal and checkpoint-drift centered kernel alignment (CKA) to identify whether vision-language prior is preserved or reshaped during fine-tuning. Second, we perform targeted attention knockouts to examine whether vision-language information is functionally utilized. Third, we combine attention localization and input editing to assess whether the resulting routing patterns reflect robust vision-language grounding or shallow visual shortcuts. Through this framework, we uncover the following key observations:

$\pi_{0.5}$ and OpenVLA exhibit distinct modality-specific adaptation dynamics during VLA fine-tuning. Specifically, our CKA analysis reveals that $\pi_{0.5}$ exhibits highly unstable layer-wise cross-modal fusion dynamics, primarily reorganizing textual representations into task-conditioned control features. In contrast, OpenVLA exhibits smoother but often weaker intermediate-layer image–text alignment, while preserving its text-pooled representations and reorganizing primarily vision-pooled and joint-pooled subspaces.

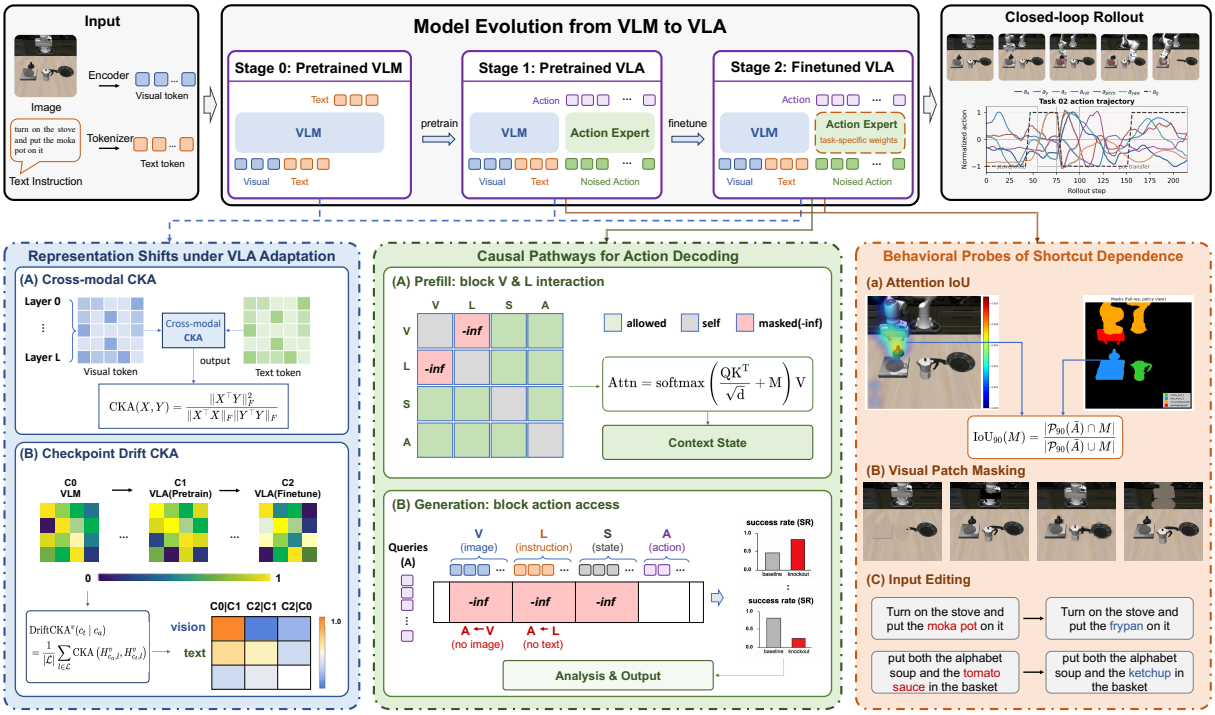

Figure 1: Overview of VLA-Trace. The framework progressively diagnoses VLA models by tracing representation dynamics, identifying causal pathways for action decoding, and probing behavioral reliance on shortcut dependence.

$\pi_{0.5}$ and OpenVLA exhibit distinct multimodal routing strategies and layer-wise dependency during action decoding. Attention knockout experiment across visual and textual pathways illustrates that $\pi_{0.5}$ is dominated by a concentrated visual-to-action routing pathway during decoding, which limits the functional role of language. Conversely, OpenVLA distributes control-relevant information across both visual and textual access.

VLA policies excel at visually grounded trajectory but remain limited in fine-grained semantic following. Attention localization and input editing reveal a gap between visual grounding and semantic following. Although policies localize to manipulation-relevant regions, they often ignore fine-grained semantic modifications, suggesting strong visually grounded trajectory imitation and weak compositional language control.

These observations highlight future directions for effective representation preservation and adaptation, causal VLA circuits, and compositional semantic control in VLA systems. Overall, this work proposes a systematic framework for probing how vision-language prior knowledge influences action control in VLA models. By identifying these bottlenecks in multimodal adaptation, causal routing, and semantic grounding, our progressive diagnosis provides guidance for developing more interpretable and robust next-generation VLA models.

2 Related Work

In this section, we review related work around two topics: *VLA models* and *mechanistic and representation analysis of VLA models*. These areas motivate the design of VLA-Trace as a progressive analysis framework.

2.1 Vision-Language-Action Models

VLA models adapt the vision-language priors of VLMs to embodied control. Unlike world foundation models (NVIDIA et al., 2025; Liao et al., 2025; Ye et al., 2026), which model environment dynamics through high-dimensional video generation, VLAs offer a more direct and lightweight paradigm by mapping multimodal representations to robot actions (Reed et al., 2022; Brohan et al., 2022; Zitkovich et al., 2023; Li et al., 2024b). Recent advances, including Octo (Team et al., 2024), OpenVLA (Kim et al., 2024), and π-style policies (Black et al., 2026; Intelligence et al., 2025b,a, 2026), have shown strong generalization across tasks and embodiments, while systems such as *Gemini Robotics* (Team et al., 2025b,a) and GR00T N1 (Bjorck et al., 2025) further extend VLA capabilities to complex manipulation and control.

Despite these advances, it remains unclear how VLM representations are reshaped during VLA adaptation and whether pretrained knowledge is preserved and functionally utilized during action generation. Prior work shows that direct VLM-to-

VLA finetuning can degrade visual and language representations (Grover et al., 2025; Hancock et al., 2025), motivating a closer examination of how representation changes influence embodied behavior.

2.2 Mechanistic Analysis of VLA Models

Recent studies investigate the underlying mechanisms of VLA models, which can be broadly categorized into three directions:

Representation-level analysis investigates the information encoded in hidden states. Probing analyses show that hidden states can encode symbolic properties, relations, action states, and latent transition information, suggesting that semantically meaningful state information is present in VLA representations (Fang et al., 2025; Hancock et al., 2025; Grant et al., 2026). **Causal mechanism analysis** moves toward mechanistic interpretability by steering activations, identifying task-relevant attention heads, and extracting sparse features that influence policy behavior (Mitra et al., 2025; Haon et al., 2025; Swann et al., 2026). **Input intervention analysis** examines grounding and robustness at input levels, revealing that VLA policies remain sensitive to distractors, background changes, object grounding errors, and failures to follow language instructions (Kerr et al., 2023; Li et al., 2024a; Hancock et al., 2024; Xie et al., 2026; Fei et al., 2025).

Nevertheless, existing analyses remain fragmented. Representation-level studies reveal what information is encoded, but not whether it is causally used during action decoding. Input intervention analyses expose behavioral failures, yet often leave the internal pathway from representation shifts to failure modes unclear. To bridge this gap, we propose **VLA-Trace**, a unified diagnostic framework that integrates representation geometry analysis, causal analysis, and input intervention into a stage-wise progressive pipeline to understand embodied control behavior.

3 Representation Analysis Framework

Overview. We investigate three stage-wise representations: C0, the pretrained VLM; C1, the pretrained VLA; and C2, the task-finetuned VLA. This staged formulation enables us to trace how general vision-language knowledge is preserved or reorganized during robotic policy learning. Built upon this progression, our framework integrates three complementary analyses. First, representation-level analysis uses CKA to examine whether visual, textual, and joint representation geometries are preserved or reshaped across three stages. Second, causal analysis (stage 2) employs attention knockouts to evaluate whether these representations are functionally required for action decoding. Third, input intervention analysis (stages 1-2), including attention localization, visual masking, and input editing, investigates whether the identified pathways lead to robust spatial grounding, shortcut dependence, or limited semantic control in behavior. Tab. 1 summarizes the probing design, while Tab. 12 presents the evaluated models and designed benchmarks, with additional results in the Appendix.

Stage	Analysis Probe	Benchmarks	Models
S1	Cross-Modal CKA *vision–language alignment*	LI/CO	P/O
	Checkpoint-Drift CKA *stage-wise drift*	LI	P/O
S2	Attention Knockout *modality reliance*	LI/CV/RT	P/O/F
	Layer-Wise Knockout *layerwise control*	LI/CV/RT	P/O/F
S3	Attention IoU *region overlap*	L10	P/O
	Visual Patch Mask *object/spatial/background*	LI/CV/RT/SP	P/O/F/X
	Input Editing *image/prompt sensitivity*	L10	P/O

Table 1: Overview of the VLA-Trace framework. Probes progress from representation geometry to causal attribution and behavioral manifestation; italic text summarizes each probe's target. Blue benchmark tags: LI =LIBERO, CO =COCO, CV =CALVIN, RT =RoboTwin2.0, SP =Simpler, L10 =LIBERO-10. Green model tags: P =$\pi_{0.5}$, O =OpenVLA, F =OFT, X =X-VLA.

Experiment Setup. We primarily focus on $\pi_{0.5}$ and OpenVLA, as they present contrasting structures that are highly diagnostic for language and vision utilization. $\pi_{0.5}$ employs a PaliGemma (Beyer et al., 2024) with an action expert under flow-matching action generation. Its transformer input consists of multi-view images, a BOS token, the task instruction, and a newline token, allowing bidirectional visual-language interaction during context formation. Conversely, OpenVLA utilizes an autoregressive Llama-based architecture(Touvron et al., 2023) where image patches are followed by a prompt template (e.g., *"In: What action should the robot take to {instruction}? Out:"*). For evaluation, we use COCO (Lin et al., 2015) as a generation-domain image-text reference and LIBERO (Liu et al., 2023) as the robot-domain benchmark.

3.1 Representation Shifts under VLA Adaptation

Evaluation Metrics. First, we investigate the evolution of latent representations across training stages to analyze the impact of robot-domain training on vision–language alignment. Specifically, we measure CKA similarity (Kornblith et al., 2019) between pooled visual and textual representations across layers and stages using COCO and LIBERO as evaluation datasets. We focus on two complementary analyses:

(1) Cross-modal CKA measures the alignment between visual and textual representations within a model. Given two representation matrices $X \in \mathbb{R}^{N \times d_x}$ and $Y \in \mathbb{R}^{N \times d_y}$, the linear CKA similarity is defined as: $\mathrm{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \|Y^\top Y\|_F}$

(2) Checkpoint Drift CKA characterizes representational changes across training stages. Specifically, for each layer, we keep three pooled views: *vision pooled*, *text pooled*, and *joint pooled*. For a given view v, we first compute the CKA between the same layer of an anchor checkpoint c_a and a target checkpoint c_t, and then average these layer-wise values: $\mathrm{DriftCKA}^v(c_t \mid c_a) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \mathrm{CKA}\Big(H^v_{c_a,l}, H^v_{c_t,l}\Big)$ where $H^v_{c,l} \in \mathbb{R}^{N \times d}$ denotes the sample-by-feature matrix of view v at layer l from checkpoint c, and $\mathcal{L}$ denotes the set of evaluated layers. Therefore, Drift CKA measures the geometric similarity of representations across different training stages. Higher values indicate stronger geometric preservation across checkpoints, whereas lower values reflect greater representational reorganization. Further experimental details are provided in the Appendix A.2.

3.1.1 CKA Alignment Analysis

Vision-Language Alignment. Fig. 2 reveals different layer-wise cross-modal organization patterns in $\pi_{0.5}$ and OpenVLA. For $\pi_{0.5}$, cross-modal CKA is broadly distributed across intermediate layers and varies substantially across checkpoints and datasets, suggesting active cross-modal reorganization during VLA pretraining and task finetuning. OpenVLA, in contrast, often shows smoother but weaker image–text CKA across intermediate layers, with high alignment frequently concentrated near the boundary or terminal layers. This observation may stem from differences in architectural design, particularly the contrast between bidirectional and unidirectional attention mechanisms. Specifically, *the bidirectional attention mechanism in $\pi_{0.5}$ may induce more complex and fluctuating cross-modal fusion dynamics, resulting in less stable alignment patterns across layers* ***(Finding 1)***.

Checkpoint Drift and Adaptation. Next, we analyze checkpoint-drift CKA to identify preserved and reorganized subspaces during robot adaptation. In Fig. 3, $\pi_{0.5}$ exhibits substantially stronger drift in textual representations, suggesting that language representations are heavily reorganized into task-conditioned control features. In contrast, OpenVLA primarily restructures visual and joint representation spaces. These results reveal *distinct modality-specific preservation and adaptation dynamics across the two models* ***(Finding 2)***.

3.2 Causal Pathways for Action Decoding

Evaluation Metrics. Second, we probe how latent representations causally influence action generation. Specifically, we decompose inference into representation formation and action decoding, and perform targeted attention knockout. This allows us to determine whether a modality contributes by shaping the initial multimodal context, by directly conditioning the action tokens during decoding, or by both mechanisms. Further experimental details are provided in the Appendix A.3.

(1) Cross-modal Interaction (Prefill). The prefill stage encodes all input modalities (vision and language) to construct the initial internal representations. To isolate the role of cross-modal interaction, we block the attention between vision and language tokens (V–L attention), thereby enforcing independent modality processing. This allows us to assess how cross-modal integration during representation formation influences action generation.

(2) Modality Dependency (Gen). During generation, actions are produced autoregressively. We perform a modality knockout by blocking attention from action tokens to vision or language tokens, isolating the dependence of action decoding on multimodal context.

This design disentangles two roles of multimodal information in VLAs: (i) cross-modal interaction during representation encoding, and (ii) modality dependency during action decoding. By intervening at these two stages, we can distinguish whether performance gains arise from better multimodal understanding or more effective utilization of modality-specific information during generation. Three attention configurations are considered: **i.** Baseline, where all modalities are provided; **ii.** No

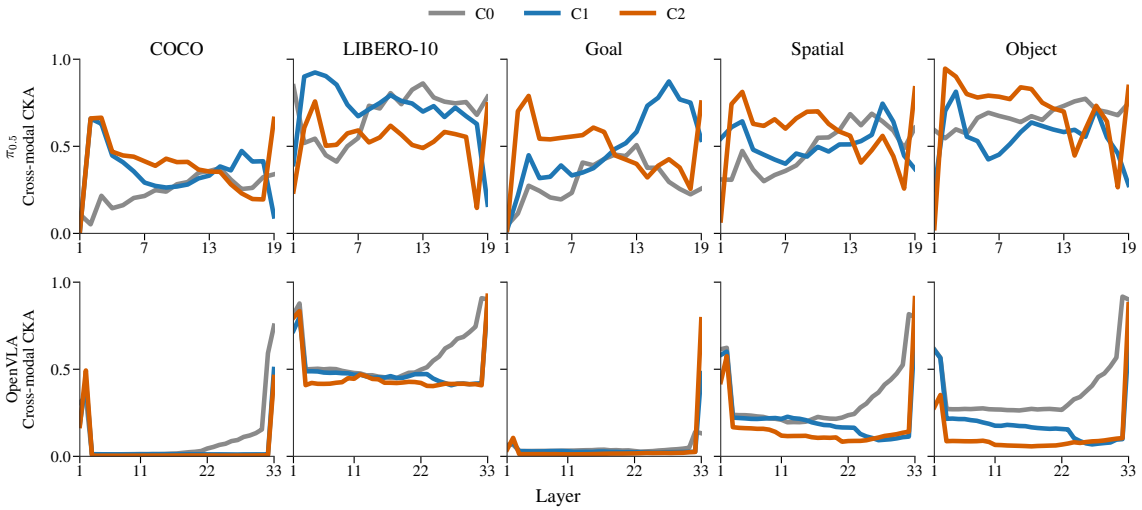

Figure 2: Layer-wise image–text CKA across datasets and training stages. The panels compare cross-modal alignment for $\pi_{0.5}$ and OpenVLA. Each curve denotes one checkpoint: C0, the pretrained VLM; C1, the pretrained VLA; and C2, the task-finetuned VLA.

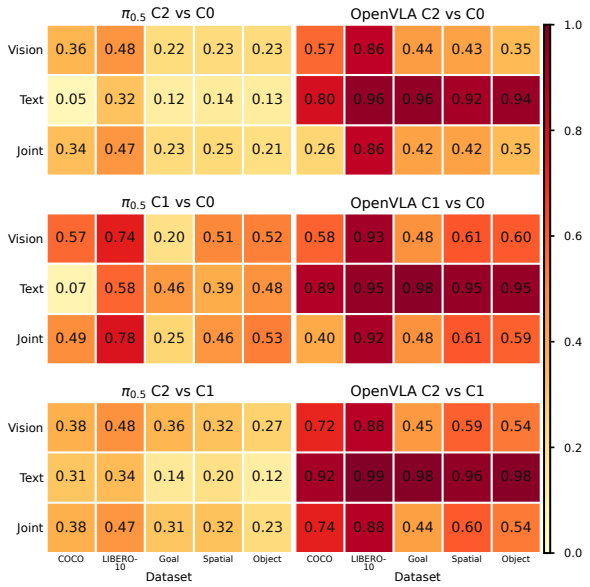

Figure 3: Stage-wise checkpoint-drift CKA for $\pi_{0.5}$ and OpenVLA. Each cell reports the mean matched-layer CKA for vision, text, and joint representations.

Model	P	G-T	G-I	LIBERO-10	Goal	Spatial	Object
$\pi_{0.5}$	✓	✓	✓	93.5	96.0	98.5	99.5
	✗	✓	✓	0.0	11.5	77.0	71.5
	✓	✗	✓	39.0	96.5	99.0	98.0
	✓	✓	✗	0.0	4.0	0.0	0.0
	✗	✗	✓	71.5	10.5	70.5	46.0
	✗	✓	✗	0.0	0.0	0.0	0.0
OpenVLA	✓	✓	✓	58.0	74.5	75.5	74.0
	✗	✓	✓	0.0	0.0	0.0	0.0
	✓	✗	✓	0.0	0.0	0.0	0.0
	✓	✓	✗	1.0	16.0	44.0	32.5
	✗	✗	✓	0.0	0.0	0.0	0.0
	✗	✓	✗	0.0	0.0	0.0	0.0

Table 2: All-layer attention knockout results. Values show success rates. Rows correspond to the original settings used throughout the paper: Baseline, Prefill, Gen: no text, Gen: no image, Comb. + gen no text, and Comb. + gen no image. Here, 'P' denotes the prefill pathway, while 'G-T' and 'G-I' denote text and image availability during generation, respectively.

Image, where visual attention is removed; **iii.** No Text, where language attention is excluded. In this experiment, model performance is evaluated using success rate (%) over 200 episodes (20 trials across 10 LIBERO-10 tasks).

3.2.1 Attention Knockout

We first apply all-layer attention knockout to measure the global necessity of each pathway, and then use layer-wise knockout to localize where the dependency is concentrated. Tab 2 reports the all-layer results across four LIBERO suites. For $\pi_{0.5}$, removing visual access during generation is consistently destructive. `Gen:no image` reduces success to nearly zero across all datasets. In contrast, `Gen:no text` largely preserves performance on Goal, Object, and Spatial, although LIBERO-10 remains more sensitive. The prefill-stage interventions further reinforce this interpretation. Moreover, combining prefill no-V–L with generation no-image completely collapses performance. These results suggest that the most fragile decoding-time pathway in $\pi_{0.5}$ is the visual-to-action route.

OpenVLA exhibits a different dependency pattern. All-layer `Gen:no text` reduces success to 0.00% on every LIBERO suite, and prefill-stage image removal is equally destructive. In contrast, `Gen:no image` is harmful but does not uniformly collapse all datasets, suggesting that OpenVLA

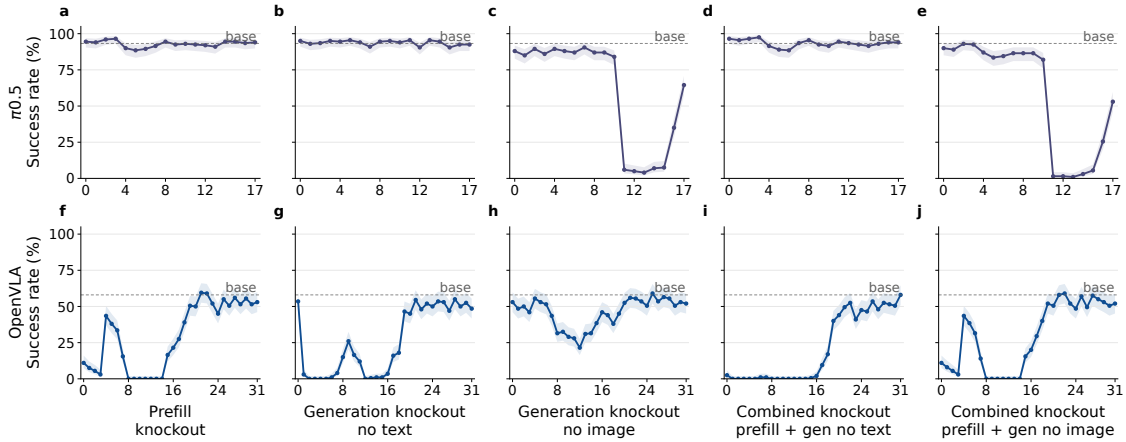

Figure 4: Attention knockout for $\pi_{0.5}$ (top) and OpenVLA (bottom) on LIBERO-10. Similar observations for LIBERO-Goal, Object, and Spatial are provided in Figure 8 and 10.

depends strongly on prompt-region access during decoding while also requiring visual grounding formed during prefill. Overall, these results indicate that *$\pi_{0.5}$ and OpenVLA exhibit distinct multimodal routing strategies during action generation: $\pi_{0.5}$ primarily depends on visual-to-action pathways, whereas OpenVLA relies on both visual and language modalities* ***(Finding 3)***.

We then use layer-wise knockout to localize these global effects. Fig. 4 shows that, on the LIBERO-10 task, $\pi_{0.5}$ has a narrow visual bottleneck. `Gen:no image` drops to 4.00% at the most vulnerable layer, while `Gen:no text` remains close to baseline across layers. Combining `prefill` with `Gen:no image` further sharpens this collapse around the same critical region. By contrast, OpenVLA shows broader vulnerable regions. `Gen:no text`, `prefill`, and their combined intervention reach 0.00% across multiple layers. The full layer-wise results over window sizes and LIBERO suites are reported in the supplementary material. These results indicate that *critical information is more broadly distributed in OpenVLA than in $\pi_{0.5}$, whereas $\pi_{0.5}$ routes action-critical visual information through a narrower bottleneck* ***(Finding 4)***.

3.3 Behavioral Probes of Grounding and Shortcut Dependence

Evaluation Metrics. Finally, we investigate whether the previously identified representation shifts and causal pathways translate into observable rollout behavior. Specifically, while Sec. 3.1 examines how multimodal representations are reorganized and Sec. 3.2 identifies the pathways required for action decoding, this section evaluates whether such pathways support robust grounding and instruction-conditioned behavior during closed-loop execution. We focus on three complementary behavioral probes:

(1) Attention IoU and Patterns. We first analyze action-to-image attention during LIBERO-10 rollouts. For each rollout step, we average action-query attention over visual patches to obtain an action-conditioned heatmap $\bar{A}$. Given a simulator-derived mask M for task-relevant objects, robot regions, or their union, we compute three localization metrics. The high-attention patch IoU measures (Jaccard, 1912) whether thresholded high-confidence attention patches overlap with task-relevant regions: $\mathrm{IoU}_{90}(M) = \frac{|\mathcal{H}_{90}(\bar{A}) \cap M|}{|\mathcal{H}_{90}(\bar{A}) \cup M|}$, where $\mathcal{H}_{90}(\bar{A}) = \{j : \bar{A}_j \geq q_{0.9}(\bar{A})\}$ denotes patches whose attention is larger than the 90th percentile. The continuous attention mass measures how much attention is allocated to the region: $\mathrm{Mass}(M) = \frac{\sum_{j \in M} \bar{A}_j}{\sum_j \bar{A}_j}$. Finally, the peak-hit rate measures whether the maximum-attention patch lies inside the target mask: $\mathrm{Hit}(M) = \mathbf{1}\left[\arg\max_j \bar{A}_j \in M\right]$. Because LIBERO-10 tasks are long-horizon instructions that typically contain two sequential subgoals, we further divide each rollout into two temporal phases by the first and second halves of the executed steps. This phase split provides an approximate but consistent proxy for the first and second instruction subgoals, allowing us to test whether visual attention evolves with the temporal structure of the task.

(2) Visual Patch Masking. We mask target objects, gripper regions, robot bodies, and backgrounds using background replacement, black masking, and mosaic masking. Comparing success rates under these perturbations reveals whether the policy depends on target-object appearance, robot-

Model	Stage	Mass	IoU_{90}		Hit
			Object	Robot+Object	
OpenVLA	Phase 1	0.5437	0.1374	0.1830	0.9204
	Phase 2	0.5563	0.1294	0.1965	0.9433
	Full	0.5882	0.1504	0.1965	0.9597
$\pi_{0.5}$	Phase 1	0.6100	0.1312	0.2303	0.6225
	Phase 2	0.5805	0.1421	0.2265	0.6259
	Full	0.6328	0.1379	0.2233	0.6349

Table 3: LIBERO-10 attention localization across temporal task phases. Metrics compare grounding on isolated target objects (`Object`) versus broader interaction regions (`Robot+Object`). `Mass` and `Hit` use the `Robot+Object` mask.

object spatial relations, or broader scene context.

(3) Input Editing. We edit either the visual layout or language instruction while preserving the task template. Image editing tests sensitivity to spatial changes, while text editing evaluates whether language can redirect policy behavior toward new targets. Failure to adapt indicates rigid dependence on the original visual-task configuration.

3.3.1 Attention IoU and Patterns

As shown in Tab. 3, both models exhibit consistent overlap between high-attention patches and Robot+Object regions. Importantly, Robot+Object IoU_{90} is consistently higher than Object-only IoU_{90} for both models, suggesting that action attention is not routed solely to semantic object regions, but to broader robot-object interaction regions that include the gripper, robot arm, manipulated object, and task-relevant affordances. Beyond spatial localization, Fig. 5 demonstrates that policies implicitly encode temporal task semantics. By splitting the two-subgoal LIBERO-10 tasks into Phase 1 and Phase 2, we observe that attention shifts consistently align with task progression. E.g., in the task "turn on the stove and put the moka pot on it," the model's focus correctly transitions from the stove to the moka pot across phases. Further experimental details are provided in Appendix A.4.

Despite both exhibiting spatial grounding, their routing mechanisms differ (as shown in Tab. 3). $\pi_{0.5}$ distributes attention broadly (higher mass, lower peak-hit rates), while OpenVLA concentrates it sparsely (higher peak-hit rates, lower mass and IoU). Corroborating Sec. 3.2, this reflects $\pi_{0.5}$'s dominant visual-to-action pathway compared to OpenVLA's reliance on both visual and prompt regions. Overall, these observations indicate that *the models exhibit spatially grounded and temporally structured action attention, routing action generation through robot-object interaction regions and shifting object-level attention across rollout phases* ***(Finding 5)***.

3.3.2 Visual Patch Masking

Next, we investigate visual perturbations. In Tab. 4, target-object masking causes the largest performance degradation in both $\pi_{0.5}$ and OpenVLA, with average drops of 76.45 and 60.25, respectively. These results indicate that target objects are not merely visually attended, but serve as causal visual anchors for action execution for both models.

Further perturbations expose model-specific visual shortcut dependencies. For $\pi_{0.5}$, masking the gripper or full robot with background replacement causes substantial performance degradation, whereas masking the robot without the gripper is much less destructive, suggesting a stronger reliance on gripper-centered interaction geometry. Conversely, OpenVLA exhibits a brittle dependence on full-robot visibility. In addition, background masking reduces average success for all models, indicating that policies exploit broader scene layouts. And further experimental details and results are provided in Appendix A.5. Together, these results reveal that *VLA policies rely on target objects, robot-object interaction region, and scene-layout cues as causal visual anchors for action execution, revealing strong visual grounding but also model-specific visual shortcut dependence* ***(Finding 6)***.

3.3.3 Input Editing

Finally, we evaluate whether the model follows fine-grained vision-language instruction variations. For text editing, Tab. 5 shows that replacing target objects in instructions does not consistently redirect policy behavior. Some edits cause large performance drops (e.g., moka pot→frypan, 75%→30%), while others have limited effect. Overall, the results suggest that *text exerts a non-negligible but limited influence on action execution, and the policy often remains biased toward the original visual-task configuration even when edited semantics dictate a behavioral shift* ***(Finding 7)***.

4 Discussion and Outlook

In this section, we summarize the main experimental findings and discuss their implications for future research directions in the VLA domain.

Representation preservation and embodied modality adaptation. First, our CKA alignment

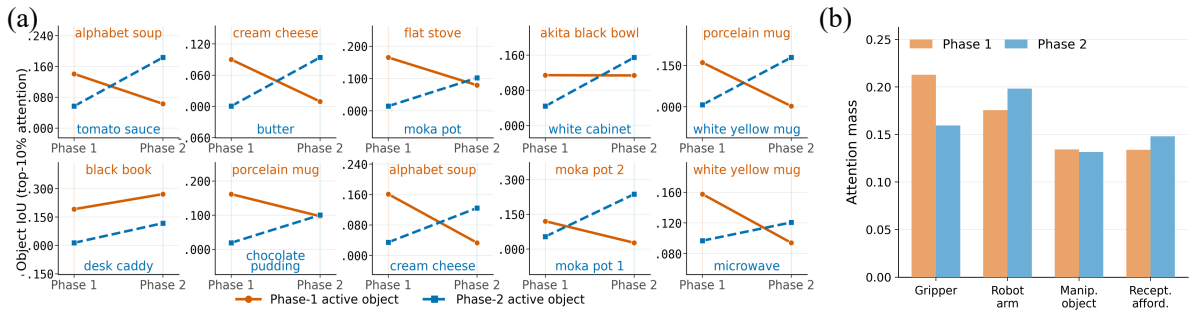

Figure 5: $\pi_{0.5}$ attention IoU and mass on LIBERO-10. (a) Object IoU dynamically shifts between the first (Phase 1) and second (Phase 2) instruction subgoals. (b) Attention mass allocation over robot and object regions. These results indicate that VLA policies successfully generate visually grounded trajectories by tracking task-relevant objects over time. See similar observations in Fig. 22 for OpenVLA and Tab. 15 for instructions.

experiment reveals that $\pi_{0.5}$ and OpenVLA exhibit distinct modality-specific adaptation dynamics at different stages. $\pi_{0.5}$ induces more fluctuating cross-modal fusion dynamics than OpenVLA, resulting in less stable alignment patterns across layers. In addition, $\pi_{0.5}$ primarily reorganizes textual representations into task-conditioned control features, whereas OpenVLA mainly restructures visual and cross-modal fusion representations.

Future VLA training should balance preserving vision-language priors with adapting them for robotic control. Insufficient adaptation limits visuomotor learning, while excessive adaptation may degrade semantic grounding and increase reliance on visual shortcuts.

Designing causal visual-language circuits for action decoding. Second, attention knockout experiments reveal distinct control dependencies: $\pi_{0.5}$ primarily relies on visual-to-action pathways, whereas OpenVLA depends on both visual grounding and prompt-region access. Layer-wise interventions further show that OpenVLA distributes control information more broadly across layers, highlighting the importance of reliable multimodal pathways for action generation.

Future VLA architectures may benefit from explicit VLA routing modules that regulate when and how visual and textual features are passed to action tokens. E.g., mid-to-late layer fusion gates could preserve high-level semantic constraints until the action generation stage, while action-conditioned cross-attention modules could dynamically select task-relevant visual regions and instruction tokens.

From visual grounding to compositional semantic control. Third, the model's attention is both spatially grounded in manipulation-relevant regions and temporally aligned with the sequential structure of the instruction. Besides, the policy often remains biased toward the original visual-task configuration even when edited semantics dictate a behavioral shift. These findings reveal a limitation in fine-grained instruction following: current VLA models can capture high-level task semantics, yet struggle to reliably incorporate subtle semantic modifications into low-level control behavior. As a result, edited instructions may fail to induce consistent compositional behavioral changes, particularly when they conflict with dominant visual or task-specific priors.

A natural future direction is to train VLA models with semantic-edit objectives, counterfactual instruction-image pairs, object-swap interventions, and contrastive behavior supervision, so that small but meaningful language changes become causally actionable in low-level control.

5 Conclusion

We present VLA-Trace, a systematic framework for analyzing representation and control in VLA models. By combining CKA, attention knockouts, attention localization, visual patch masking, and input editing, VLA-Trace connects representation dynamics with causal action dependencies and rollout behavior. Experiments on $\pi_{0.5}$ and OpenVLA reveal divergent adaptation strategies and a shared limitation: semantic control remains partial and architecture-dependent despite strong benchmark performance. These findings suggest that future VLA models should better preserve visual, language, and cross-modal semantic pathways, and integrate them more reliably into action decoding for robust compositional control.

Limitations

This study has several limitations. First, attention knockout is a controlled necessity probe, and its induced failures should not be directly equated with natural out-of-distribution failures. It reveals which pathways are required under a specific intervention design, but it does not define all forms of robustness. Second, CKA measures sample-level representation geometry and should not be interpreted as an absolute score of grounding, perception, or language understanding. Its values depend on the selected dataset, pooling strategy, and checkpoint comparison; therefore, our claims emphasize relative layer profiles and checkpoint drift rather than absolute CKA magnitudes. Third, the current analysis does not include random-feature or permutation CKA baselines, and the rollout success rates should be complemented with confidence intervals in a final submission. Fourth, attention IoU depends on simulator masks and patch resolution, so it is best interpreted as a conservative localization measure rather than a complete account of visual grounding. Fifth, VLM-style grounding probes can be confounded by whether a VLA checkpoint still supports natural language localization outputs; degraded text generation can reflect output-format mismatch as well as grounding degradation. Sixth, $\pi_{0.5}$ C2 is evaluated as a shared LIBERO-finetuned checkpoint across multiple suites, so suite-level differences reflect evaluation distribution effects rather than separate per-suite weight changes. Finally, this work focuses its complete evidence chain on $\pi_{0.5}$ and OpenVLA; our broader implementation protocol defines extensions to X-VLA, OpenVLA-OFT, Calvin, SimplerEnv, LIBERO-Plus/Pro, and RoboTwin, but future work should complete these settings before making broader claims across the full VLA design space.

Ethics Statement

This work analyzes existing VLA models using publicly available models and evaluation resources, without collecting personal data or involving human-subject studies. While our framework is intended to improve transparency and diagnosis, the analyzed models may still inherit biases, safety risks, and failure modes from their training data and design; therefore, real-world deployment requires careful evaluation, human oversight, and appropriate safety constraints.

References

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. Paligemma: A versatile 3b vlm for transfer. *Preprint*, arXiv:2407.07726.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, and 1 others. 2025. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, and 5 others. 2026. π_0: A vision-language-action flow model for general robot control. *Preprint*, arXiv:2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, and 1 others. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

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

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, and 1 others. 2025. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*.

Bryce Grant, Xijia Zhao, and Peng Wang. 2026. Not all features are created equal: A mechanistic study of vision-language-action models. In *International Conference on Learning Representations*.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I. Christensen, Hao Su, and Xuanlin Li. 2025. Enhancing generalization in vision-language-action models by preserving pretrained representations. *arXiv preprint arXiv:2509.11417*.

Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. 2024. Run-time observation interventions make vision-language-action models more visually robust. *arXiv preprint arXiv:2410.01971*.

Asher J Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. 2025. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. *arXiv preprint arXiv:2509.22195*.

Bear Haon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. 2025. Mechanistic interpretability for steering vision-language-action models. *arXiv preprint arXiv:2509.00328*.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, and 69 others. 2026. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *Preprint*, arXiv:2604.15483.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, and 37 others. 2025a. $\pi^*_{0.6}$: a vla that learns from experience. *Preprint*, arXiv:2511.14759.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, and 1 others. 2025b. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. *New Phytologist*.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*.

Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia Sycara, and Simon Stepputtis. 2024a. Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10527–10534. IEEE.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, and 1 others. 2024b. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, volume 2024, pages 26703–26721.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. 2025. Genie envisioner: A unified world foundation platform for robotic manipulation. *Preprint*, arXiv:2508.05635.

Tsung-Yi Lin, Michael Maire, Serge Belongie, and et al. 2015. Microsoft coco: Common objects in context. *Preprint*, arXiv:1405.0312.

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

Chancharik Mitra, Yusen Luo, Raj Saravanan, Dantong Niu, Anirudh Pai, Jesse Thomason, Trevor Darrell, Abrar Anwar, Deva Ramanan, and Roei Herzig. 2025. Mechanistic finetuning of vision-language-action models via few-shot demonstrations. *arXiv preprint arXiv:2511.22697*.

NVIDIA, :, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, and 60 others. 2025. Cosmos world foundation model platform for physical ai. *Preprint*, arXiv:2501.03575.

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

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, and 1 others. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.

Aiden Swann, Lachlain McGranahan, Hugo Buurmeijer, Monroe Kennedy III, and Mac Schwager. 2026. Sparse autoencoders reveal interpretable and steerable features in vla models. *arXiv preprint arXiv:2603.19183*.

Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, and 1 others. 2025a. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, and 1 others. 2025b. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, and 1 others. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2023. Florence-2: Advancing a unified representation for a variety of vision tasks. *Preprint*, arXiv:2311.06242.

Yuhan Xie, Yuping Yan, Yunqi Zhao, Handing Wang, and Yaochu Jin. 2026. Strong-vla: Decoupled robustness learning for vision-language-action models under multimodal perturbations. *arXiv e-prints*, pages arXiv–2604.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, and 17 others. 2026. World action models are zero-shot policies. *Preprint*, arXiv:2602.15922.

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

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR.

A Implementation Protocol Details

A.1 Model Interfaces and Input Templates

The VLA-Trace protocol tracks several VLA model families, including $\pi_{0.5}$, X-VLA, OpenVLA, and OpenVLA-OFT. The main paper focuses on $\pi_{0.5}$ and OpenVLA because they instantiate two distinct action-generation interfaces. $\pi_{0.5}$ builds on a PaliGemma-style VLM and uses an action expert with flow-matching action generation. In the evaluated configuration, the transformer input contains visual tokens from the image observation, a BOS token, the task instruction, and a newline token. Although the original model interface includes proprioceptive state fields, the analyzed representation stream does not tokenize proprioceptive state into the transformer prompt. OpenVLA instead formulates action prediction autoregressively over action tokens, using image patches followed by a prompt template of the form *"In: What action should the robot take to {instruction}? Out:"*. This interface distinction is important for rollout and attention-knockout experiments, where semantic instruction tokens must be separated from structural tokens such as BOS, newline markers, and system prompts. OpenVLA-OFT extends OpenVLA by replacing autoregressive action-token generation with continuous action regression during fine-tuning. This modification changes the action prediction interface and reduces dependence on sequential token decoding, making OpenVLA-OFT a useful supplementary setting for comparing how decoding paradigms influence representation dynamics and causal pathways.

A.2 CKA Protocol

Prompt templates for CKA probes. For representation CKA, we use separate neutral text templates rather than the action-generation prompts used during rollout. This choice makes the open-domain and robot-domain probes share a comparable image-description form and avoids conflating representational alignment with action-query formatting. For COCO, captions are rendered as `A photo of <caption>`. For LIBERO, task instructions are rendered as `A photo of a robot <instruction>`, where the instruction is inserted after the word "robot". For OpenVLA, the same neutral textual content is wrapped with the model-required input format, yielding *"In: ... Out:"*; for $\pi_{0.5}$, the neutral text is used directly, with the PaliGemma loader adding the required image token internally. These templates are used only for the representation-alignment probe and should not be confused with the action-generation prompt templates used by $\pi_{0.5}$ or OpenVLA during policy rollout and knockout experiments.

Pooled token views. For each sample and transformer layer, we extract hidden states over the full image–text sequence and partition them into visual and textual spans according to the model-specific token layout. Let $h_{i,t}^{\ell}$ denote the hidden state of token t at layer ℓ for sample i, and let $\mathcal{I}$ and $\mathcal{T}$ denote the image-token and text-token index sets. We define three pooled views:

$$v_i^{\ell} = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} h_{i,t}^{\ell},$$

$$q_i^{\ell} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} h_{i,t}^{\ell},$$

$$j_i^{\ell} = \frac{1}{|\mathcal{I}| + |\mathcal{T}|} \sum_{t \in \mathcal{I} \cup \mathcal{T}} h_{i,t}^{\ell}.$$

We refer to these vectors as `vision_pooled`, `text_pooled`, and `joint_pooled`, respectively. Stacking them over all probe samples gives layer-wise representation matrices $V^{\ell}, Q^{\ell}, J^{\ell} \in \mathbb{R}^{N \times d}$, where N is the number of probe examples and d is the hidden dimension. For OpenVLA, the image span corresponds to the image-patch tokens and the text span corresponds to the subsequent textual prompt tokens. For PaliGemma-style $\pi_{0.5}$ C0 extraction, the text span skips the BOS token following the image tokens.

CKA computation. We compute linear CKA over sample-wise pooled representations. Given two representation matrices $X, Y \in \mathbb{R}^{N \times d}$, we first center them along the sample dimension and compute

$$\mathrm{CKA}(X, Y) = \frac{\|X_c^{\top} Y_c\|_F^2}{\|X_c^{\top} X_c\|_F \, \|Y_c^{\top} Y_c\|_F}.$$

For layer-wise image–text alignment, we compute $\mathrm{CKA}(V^{\ell}, Q^{\ell})$ independently at each layer ℓ within the same checkpoint. For checkpoint-drift analysis, we compare matched layers across two checkpoints for each pooled view:

$$\mathrm{CKA}(V_a^{\ell}, V_b^{\ell}), \quad \mathrm{CKA}(Q_a^{\ell}, Q_b^{\ell}), \quad \mathrm{CKA}(J_a^{\ell}, J_b^{\ell}),$$

and report the mean diagonal CKA over layers. Thus, high checkpoint-drift CKA indicates preservation of the corresponding representation subspace, whereas low CKA indicates stronger representational reorganization.

A.3 Attention Knockout Protocol

We use attention knockout to test whether specific token pathways are causally required for action generation. All interventions are implemented by adding a large negative mask to selected attention logits before the softmax, while keeping model weights, input observations, and decoded action heads unchanged. For an attention layer with queries Q, keys K, values V, and hidden dimension d, the modified attention is

$$\mathrm{Attn}(Q, K, V; M) = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d}} + M\right) V, \tag{1}$$

where $M_{ij} = 0$ preserves attention from query token i to key token j, and $M_{ij} = -\infty$ blocks that attention edge.

Intervention granularity. We evaluate three levels of knockout. First, *all-layer knockout* applies the same source-blocking rule to every transformer layer, measuring the global necessity of a pathway. Second, *single-layer knockout* applies the mask only at one target layer, measuring whether the dependency is localized. Third, to reduce the possibility that information bypasses a single blocked layer through residual connections, we use *centered window knockout*. For a target layer ℓ and window size w, $w = 3$ blocks layers $\ell - 1$ to $\ell + 1$, $w = 5$ blocks $\ell - 2$ to $\ell + 2$, and $w = 7$ blocks $\ell - 3$ to $\ell + 3$. We use these windows to distinguish distributed pathways from localized bottlenecks: if all-layer removal is destructive but every local window is mild, the pathway is likely distributed; if a narrow window causes a large drop, the corresponding layer range is treated as a candidate causal bottleneck.

Token partitions. The knockout masks are defined over role-specific token sets. Let $\mathcal{V}$ denote visual tokens, $\mathcal{L}$ semantic language tokens, $\mathcal{S}$ structural prompt tokens, and $\mathcal{A}$ action-query or action-token positions. For $\pi_{0.5}$, $\mathcal{V}$ contains visual tokens from the agent and wrist views, $\mathcal{L}$ contains the task-instruction tokens, $\mathcal{S}$ contains BOS and newline tokens, and $\mathcal{A}$ contains action-query or action-slot tokens used by the action expert. For OpenVLA and OpenVLA-OFT, $\mathcal{V}$ contains image patch tokens, $\mathcal{L}$ contains the natural-language task span, $\mathcal{S}$ contains system prompt and special tokens such as *"In:"*, *"Out:"*, question text, newline, BOS, and EOS where applicable, and $\mathcal{A}$ contains autoregressive action tokens.

Prefill-stage knockout. We separate inference into a `prefill` stage and a `generation` stage. The `prefill` stage builds the internal multimodal context before action decoding. For $\pi_{0.5}$, whose VLM backbone uses bidirectional attention, we test N0 V–L prefill blocking by preventing visual and language tokens from directly attending to each other. With token groups ordered as $[\mathcal{L}; \mathcal{V}; \mathcal{A}]$, this can be represented as

$$M^{\ell}_{\text{prefill:no-V-L}} = \begin{bmatrix} \mathbf{LL} & -\infty & \mathbf{LA} \\ -\infty & \mathbf{VV} & \mathbf{VA} \\ \mathbf{AL} & \mathbf{AV} & \mathbf{AA} \end{bmatrix}, \quad (2)$$

where unmasked blocks preserve the original attention pattern and $-\infty$ blocks suppress direct vision–language exchange. For OpenVLA-style autoregressive models, the main `prefill` intervention removes image access during context construction, testing whether visual grounding must be established before action-token decoding.

Generation-stage knockout. In the generation stage, action tokens are decoded from the prefilled multimodal context. We perform modality knockout by blocking attention from action positions to selected source tokens, thereby preventing the decoder from conditioning on a specific modality during action prediction. This intervention measures the extent to which generation depends on visual tokens, semantic instruction tokens, structural prompt tokens, or their combinations.

Let the token groups be ordered as $[\mathcal{L}; \mathcal{V}; \mathcal{A}]$, where $\mathcal{L}$ denotes language tokens, $\mathcal{V}$ denotes visual tokens, and $\mathcal{A}$ denotes action tokens. For example, the generation-time no-text intervention blocks action-to-language attention while preserving the remaining attention structure:

$$M^{\ell}_{\text{gen:no-text}} = \begin{bmatrix} \mathbf{LL} & \mathbf{LV} & \mathbf{LA} \\ \mathbf{VL} & \mathbf{VV} & \mathbf{VA} \\ -\infty & \mathbf{AV} & \mathbf{AA} \end{bmatrix}. \quad (3)$$

Here, the $-\infty$ block suppresses attention from action queries to language keys. Analogously, the no-image intervention replaces the **AV** block with $-\infty$, blocking action-to-vision attention:

$$M^{\ell}_{\text{gen:no-image}} = \begin{bmatrix} \mathbf{LL} & \mathbf{LV} & \mathbf{LA} \\ \mathbf{VL} & \mathbf{VV} & \mathbf{VA} \\ \mathbf{AL} & -\infty & \mathbf{AA} \end{bmatrix}. \quad (4)$$

For autoregressive OpenVLA-style models, these knockout masks are applied on top of the original causal mask, so they only remove source positions that would otherwise be visible to the current action token.

Additional experimental results. Beyond the all-layer knockout results reported in the main paper, we provide full layer-wise (Tab. 6) and windowed knockout sweeps to assess whether each dependency is distributed across the network or concentrated in a narrow layer range. We use all-layer knockout to establish global pathway necessity, layer-wise and centered-window knockout to localize vulnerable regions, and semantic/structural token splits to avoid conflating task-language grounding with dependence on prompt-formatting tokens. Fig. 6–8 report $\pi_{0.5}$ the results of the four LIBERO suites under 3-, 5-, and 7-layer windows; Figures 9–10 report OpenVLA's results; Fig. 11–13 extend the analysis to OpenVLA-OFT; and Fig. 14–15 show OpenVLA-OFT results on RoboTwin tasks. Together, these results provide consistent evidence about whether action generation depends primarily on visual tokens, semantic language tokens, structural prompt tokens, prefill-stage visual grounding, or localized layer bottlenecks across model families and evaluation suites.

A.4 Attention Localization Metrics and Visualizations

We complement the knockout analysis with attention-localization diagnostics that measure where action-relevant attention is spatially allocated during policy execution. All localization metrics are computed on simulator-aligned image patch grids. For each rollout step, we extract attention from action-query or action-token positions to image patches and average over action positions to obtain an action-conditioned heatmap $\bar{A}$. Let $\bar{A}_j$ denote the attention assigned to image patch j, and let M denote a simulator-derived binary mask for task-relevant objects, gripper/robot regions, or their union. We report three complementary localization metrics.

First, continuous attention mass measures the fraction of action attention assigned to the target region:

$$\text{Mass}(M) = \frac{\sum_{j \in M} \bar{A}_j}{\sum_j \bar{A}_j}. \quad (5)$$

Second, to evaluate whether high-confidence attention regions spatially overlap with task-relevant regions, we compute a 90th-percentile thresholded

attention IoU. Let

$$\tau_{90} = q_{0.9}(\bar{A}), \tag{6}$$

where $q_{0.9}(\bar{A})$ denotes the 90th percentile of the heatmap values. We define the high-attention patch set as

$$\mathcal{H}_{90}(\bar{A}) = \{j : \bar{A}_j \geq \tau_{90}\}. \tag{7}$$

The thresholded IoU is then

$$\mathrm{IoU}_{90}(M) = \frac{|\mathcal{H}_{90}(\bar{A}) \cap M|}{|\mathcal{H}_{90}(\bar{A}) \cup M|}. \tag{8}$$

Third, peak-hit rate measures whether the single maximum-attention patch falls inside the target mask:

$$\mathrm{Hit}(M) = \mathbf{1}\left[\arg\max_j \bar{A}_j \in M\right]. \tag{9}$$

These metrics capture different aspects of visual grounding: Mass measures total attention allocation to a region, IoU_{90} measures spatial overlap between the target mask and attention regions above the 90th-percentile threshold, and Hit measures whether the strongest attended patch is task-relevant.

Temporal phase split. LIBERO-10 tasks often involve long-horizon instructions with multiple sequential subgoals. To analyze whether visual attention evolves with the temporal structure of the task, we divide each rollout into two phases using the first and second halves of the executed trajectory. This split provides a simple and consistent proxy for early versus late subgoals without requiring manual step-level annotation. We compute all localization metrics separately for each phase and for each mask type.

IoU visualization schematic. To make the localization metrics visually interpretable, we include a schematic visualization of the IoU computation in Fig. 16. The action-conditioned attention map is first projected onto the image patch grid. We then threshold the map at the 90th percentile to obtain the high-attention patch set $\mathcal{H}_{90}(\bar{A})$ and compare it with the simulator-derived target mask M. The overlap region contributes to the numerator of IoU_{90}, while the union of high-attention and target patches forms the denominator.

Qualitative action-to-image attention. In addition to the quantitative localization metrics, we visualize action-conditioned image attention across execution stages. Fig. 17 compares pretrained and fine-tuned $\pi_{0.5}$ and OpenVLA checkpoints over early, middle, and final rollout steps. And Figures 20 and 21 provide more qualitative attention visualizations, showing the temporal evolution of action-to-text attention and the layer-wise redistribution of modality attention before and after fine-tuning. These visualizations are intended as qualitative complements to the mask-based metrics, which show how action attention moves over the scene during manipulation, while the IoU, mass, and hit-rate metrics quantify whether this attention overlaps with simulator-derived task regions.

Token-wise text-to-image attention. We also visualize text-to-image attention to inspect how individual instruction tokens attend to visual patches. For each displayed instruction token, we project its attention over image patches back to the image grid. Fig. 18 and 19 compare pretrained and fine-tuned checkpoints for $\pi_{0.5}$ and OpenVLA, respectively. These plots help separate action-conditioned visual grounding from token-level language-to-image grounding: action-to-image attention measures which visual regions are used for decoding actions, whereas text-to-image attention shows how instruction tokens are visually grounded during representation formation.

A.5 Visual Perturbation and Editing Protocol

We use visual perturbation to test whether the image regions highlighted by attention analyses are also causally required for action execution. For each evaluation episode, we construct region masks over the input observation and replace selected regions before feeding the edited image to the policy. Model weights, task instructions, and the rollout environment are otherwise unchanged. We evaluate four primary region types: target objects, gripper regions, robot body regions, and background. Target-object masks are derived from task-relevant object annotations or simulator-aligned segmentation masks. Gripper masks cover the end-effector region, robot-body masks cover the visible robot morphology, and full-robot masks combine gripper and body regions. Background masks cover image regions outside the annotated task objects and robot regions. When multiple camera views are used, the same perturbation type is applied to

the corresponding visible regions in each view.

Mask replacement styles. We consider multiple replacement styles because they remove different visual cues. In *background-color replacement*, pixels inside the selected mask are replaced by a local or estimated background color. This removes object appearance and weakens boundary cues, making it the most aggressive test of whether the policy depends on the removed region. In *black masking*, pixels are replaced by a uniform black value. This removes texture and color but can preserve a coarse silhouette through the artificial mask boundary. In *mosaic masking*, the selected region is replaced by a low-resolution or block-wise mosaic pattern. This disrupts fine-grained appearance while preserving partial spatial occupancy and coarse geometry. Comparing these styles helps distinguish dependence on visual identity, object geometry, gripper-object spatial relations, and broader scene-layout cues.

Region-specific perturbations. Target-object masking tests whether the manipulated object is a causal visual anchor rather than merely an attended region. Gripper masking tests whether the policy depends on end-effector pose and local contact geometry. Robot-body and full-robot masking test whether policies use broader robot configuration cues beyond the gripper. Background masking tests whether policies rely on global scene layout, support surfaces, or contextual correlations outside explicitly task-relevant objects. Because these regions are not semantically equivalent, we interpret performance drops relative to the masked region and replacement style rather than treating all perturbations as generic image corruption.

Additional experimental results. The main paper reports averaged perturbation effects across masking strategies. In the supplement, we provide task-level success rates for all evaluated model families, including $\pi_{0.5}$, OpenVLA, OpenVLA-OFT, and X-VLA, to show how visual perturbation sensitivity varies across architectures and task distributions. Tab. 7 reports results on LIBERO-10, Tab. 8 reports results on LIBERO-Object, Tab. 9 reports results on LIBERO-Spatial, and Tab. 10 reports results on LIBERO-Goal. We additionally report X-VLA perturbation results on the Simpler benchmark in Tab. 11, providing an out-of-LIBERO check of the same masking protocol. Together, these tables provide a task-level view of how target-object, gripper, robot, and background perturbations affect different VLA architectures across the LIBERO benchmark suites.

B Instruction

We provide the full set of task instructions used in our benchmark evaluation to make the experimental coverage explicit and reproducible. Tab. 15 summarizes the instructions for LIBERO, RoboTwin2.0, and SimplerEnv-BridgeV2. The LIBERO suites cover complementary manipulation capabilities: LIBERO-10 evaluates long-horizon compositional manipulation, LIBERO-Object focuses on object-centric understanding, LIBERO-Spatial probes spatial reasoning, and LIBERO-Goal evaluates goal-conditioned control. For RoboTwin2.0, we evaluate a subset of 5 out of 50 tasks, focusing on representative bimanual manipulation scenarios. SimplerEnv-BridgeV2 evaluates real-to-sim transfer on Bridge-style manipulation tasks. Tab. 16 separately lists the CALVIN tasks, which cover sequential multi-skill manipulation, including object rotation, pushing, grasping, lifting, stacking, container interaction, drawer operation, and light control.

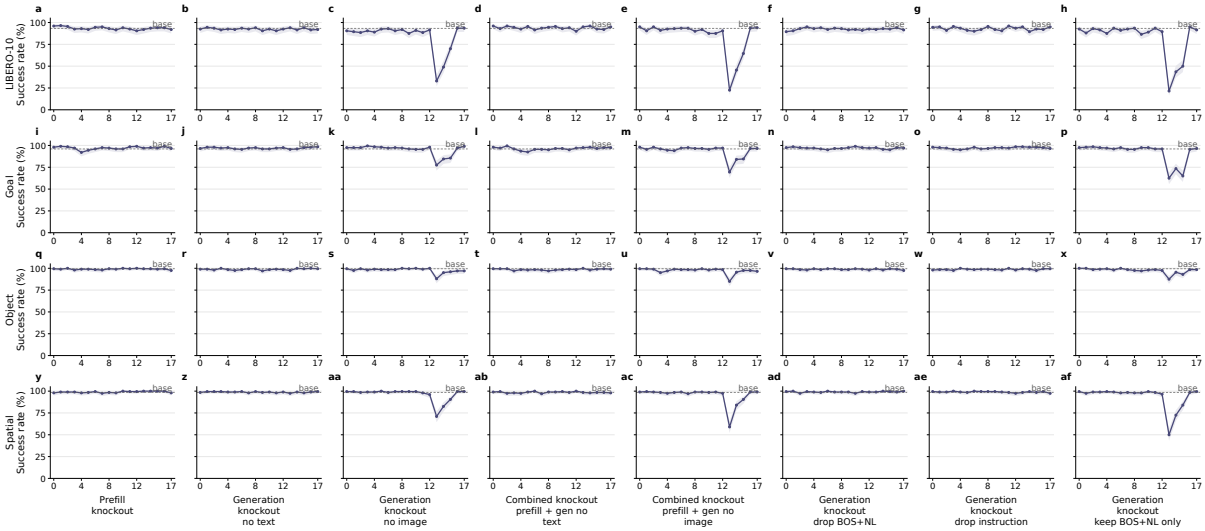

Figure 6: Layer-wise knockout results for $\pi_{0.5}$ on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **3-layer** knockout window centered at the indicated layer.

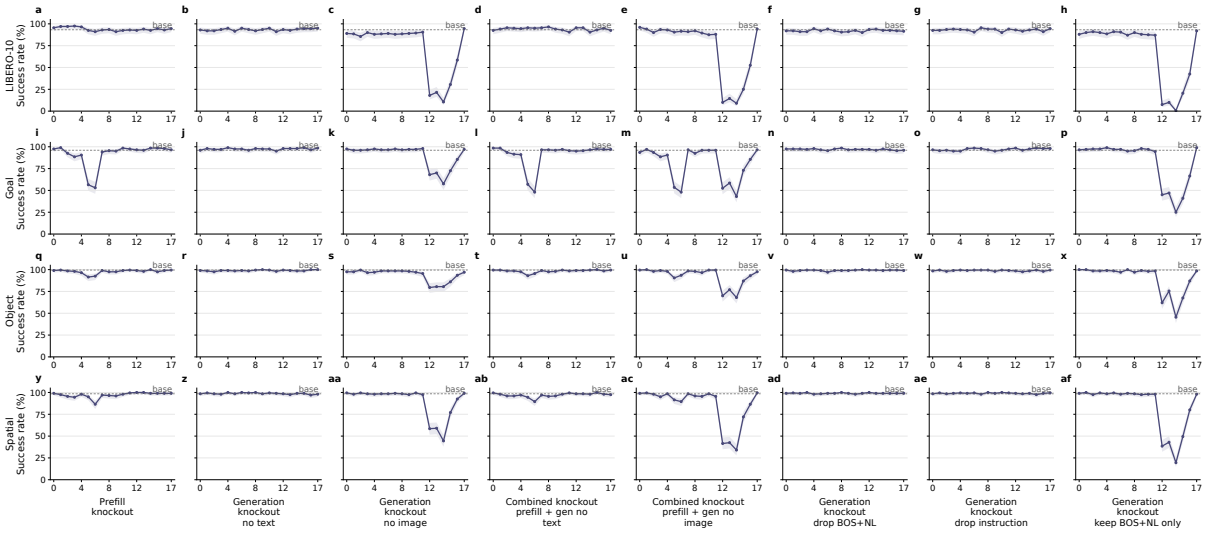

Figure 7: Layer-wise knockout results for $\pi_{0.5}$ on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **5-layer** knockout window centered at the indicated layer.

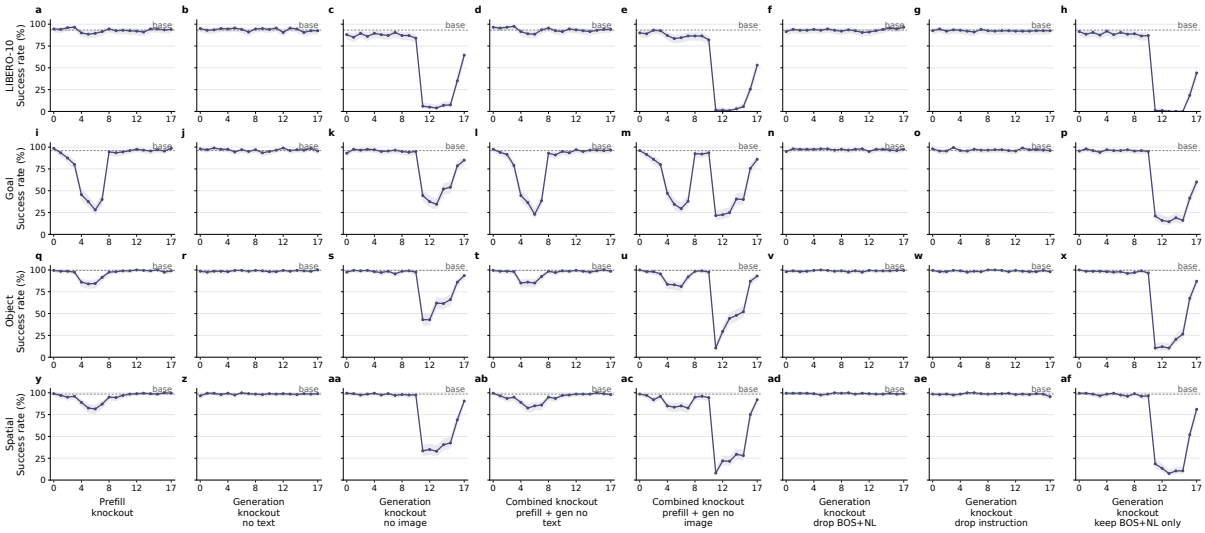

Figure 8: Layer-wise knockout results for $\pi_{0.5}$ on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **7-layer** knockout window centered at the indicated layer.

Model	Setting	LIBERO-10	LIBERO-Object	LIBERO-Spatial	LIBERO-Goal	Avg. Drop
$\pi_{0.5}$	Baseline	75.00	95.60	95.80	80.00	-
	Target (BG)	0.00	13.00	7.80	19.80	**76.45**
	Target (Black)	19.60	61.60	23.00	58.80	45.85
	Target (Mosaic)	47.20	74.00	50.20	58.40	29.15
	Gripper (BG)	8.40	3.20	64.00	59.20	52.90
	Gripper (Black)	52.60	97.40	94.80	74.80	6.70
	Gripper (Mosaic)	53.20	79.40	89.80	74.20	12.45
	Robot (BG)	3.40	26.40	47.60	44.00	56.25
	Robot (Black)	30.20	93.60	95.00	69.80	14.45
	Robot (Mosaic)	39.20	74.60	89.60	68.60	18.60
	Robot w/o Gripper (BG)	52.60	91.80	97.40	76.80	6.95
	Robot w/o Gripper (Black)	58.40	92.80	99.00	76.40	4.85
	Robot w/o Gripper (Mosaic)	64.40	96.60	96.40	79.00	2.50
	Background (Black)	41.00	79.40	67.20	57.80	25.25
	Background (Mosaic)	57.80	86.60	93.00	78.00	7.75
OpenVLA	Baseline	54.33	70.00	79.67	74.00	-
	Target (BG)	5.00	0.33	36.67	45.00	47.75
	Target (Black)	20.67	55.67	63.33	73.00	16.33
	Target (Mosaic)	25.00	59.67	70.00	67.67	13.92
	Gripper (BG)	10.67	1.00	19.00	38.00	52.33
	Gripper (Black)	31.33	72.00	70.67	69.33	8.67
	Gripper (Mosaic)	20.33	3.67	53.00	63.00	34.50
	Robot (BG)	0.00	0.00	0.00	0.33	**69.42**
	Robot (Black)	0.67	4.00	3.33	6.00	66.00
	Robot (Mosaic)	0.33	0.00	2.00	8.00	66.92
	Robot w/o Gripper (BG)	10.00	34.33	22.00	25.67	46.50
	Robot w/o Gripper (Black)	11.67	17.33	31.67	26.00	47.83
	Robot w/o Gripper (Mosaic)	16.00	15.67	49.67	38.33	39.58
	Background (Black)	26.33	42.67	47.33	41.67	30.00
	Background (Mosaic)	48.00	65.67	82.33	70.67	2.83
OpenVLA-OFT	Baseline	94.80	99.80	92.80	97.40	-
	Target (BG)	10.40	57.00	26.80	49.60	**60.25**
	Target (Black)	15.20	84.80	35.40	60.20	47.30
	Target (Mosaic)	48.20	98.40	76.00	88.40	18.45
	Gripper (BG)	89.80	94.20	92.40	96.00	3.10
	Gripper (Black)	89.80	98.80	92.80	97.00	1.60
	Gripper (Mosaic)	95.00	100.00	93.80	97.60	-0.40
	Robot (BG)	66.80	96.00	81.40	67.20	18.35
	Robot (Black)	79.00	96.80	86.80	87.20	8.75
	Robot (Mosaic)	84.40	97.60	91.60	94.60	4.15
	Robot w/o Gripper (BG)	85.40	98.60	90.40	90.60	4.95
	Robot w/o Gripper (Black)	84.00	99.80	91.00	92.60	4.35
	Robot w/o Gripper (Mosaic)	91.20	98.80	91.20	96.40	1.80
	Background (Black)	73.80	47.40	83.20	72.20	27.05
	Background (Mosaic)	84.60	95.00	88.20	97.60	4.85

Table 4: Average success rates (%) across LIBERO datasets for $\pi_{0.5}$, OpenVLA, and OpenVLA-OFT under each masking strategy.

Original Prompt	**Orig.**	**Edit**	**Edited Prompt**
put both the alphabet soup and the tomato sauce in the basket	55	60	put both the alphabet soup and the ketchup in the basket
put both the cream cheese box and the butter in the basket	100	90	put both the cream cheese box and the milk in the basket
turn on the stove and put the moka pot on it	75	30	turn on the stove and put the frypan on it
put the black bowl in the bottom drawer of the cabinet and close it	90	30	put the white bottle in the bottom drawer of the cabinet and close it
put the white mug on the left plate and put the yellow and white mug on the right plate	80	70	put the red coffee mug on the left plate and put the yellow coffee mug on the right plate
pick up the book and place it in the back compartment of the caddy	90	30	pick up the mug and place it in the back compartment of the caddy
put the white mug on the plate and put the chocolate pudding to the right of the plate	75	100	put the red mug on the plate and put the chocolate pudding to the right of the plate

Table 5: Success rate (%) comparison between original and edited prompts.

Model	Setting	LIBERO-10	Goal	Spatial	Object	Avg.
$\pi_{0.5}$	Baseline	93.5 (0.0)	96.0 (0.0)	98.5 (0.0)	99.5 (0.0)	96.9 (0.0)
	Gen: drop BOS+newline	57.5 (-36.0)	97.5 (+1.5)	99.0 (+0.5)	99.5 (0.0)	88.4 (-8.5)
	Gen: drop instruction	86.0 (-7.5)	97.0 (+1.0)	100.0 (+1.5)	99.5 (0.0)	95.6 (-1.3)
	Gen: keep BOS+newline only	0.0 (-93.5)	1.5 (-94.5)	0.0 (-98.5)	0.0 (-99.5)	0.4 (-96.5)
	Gen: no image	0.0 (-93.5)	4.0 (-92.0)	0.0 (-98.5)	0.0 (-99.5)	1.0 (-95.9)
	Gen: no text	39.0 (-54.5)	96.5 (+0.5)	99.0 (+0.5)	98.0 (-1.5)	83.1 (-13.8)
	Prefill	0.0 (-93.5)	11.5 (-84.5)	77.0 (-21.5)	71.5 (-28.0)	40.0 (-56.9)
	Comb. + gen no image	0.0 (-93.5)	0.0 (-96.0)	0.0 (-98.5)	0.0 (-99.5)	0.0 (-96.9)
	Comb. + gen no text	71.5 (-22.0)	10.5 (-85.5)	70.5 (-28.0)	46.0 (-53.5)	49.6 (-47.3)
OpenVLA	Baseline	58.0 (0.0)	74.5 (0.0)	75.5 (0.0)	74.0 (0.0)	70.5 (0.0)
	Gen: drop newline	53.5 (-4.5)	75.5 (+1.0)	80.0 (+4.5)	71.5 (-2.5)	70.1 (-0.4)
	Gen: drop prompt, keep newline	0.0 (-58.0)	0.0 (-74.5)	0.0 (-75.5)	0.0 (-74.0)	0.0 (-70.5)
	Gen: no image	1.0 (-57.0)	16.0 (-58.5)	44.0 (-31.5)	32.5 (-41.5)	23.4 (-47.1)
	Gen: no text	0.0 (-58.0)	0.0 (-74.5)	0.0 (-75.5)	0.0 (-74.0)	0.0 (-70.5)
	Prefill	0.0 (-58.0)	0.0 (-74.5)	0.0 (-75.5)	0.0 (-74.0)	0.0 (-70.5)
	Comb. + gen no image	0.0 (-58.0)	0.0 (-74.5)	0.0 (-75.5)	0.0 (-74.0)	0.0 (-70.5)
	Comb. + gen no text	0.0 (-58.0)	0.0 (-74.5)	0.0 (-75.5)	0.0 (-74.0)	0.0 (-70.5)

Table 6: All-layer attention knockout results across LIBERO suites. Each cell reports success rate (%) and the absolute change relative to the suite-specific all-layer baseline in parentheses. For $\pi_{0.5}$, `Prefill` blocks V–L attention; for OpenVLA, `Prefill` removes image access. `Comb.` combines the corresponding prefill intervention with the specified generation-stage intervention.

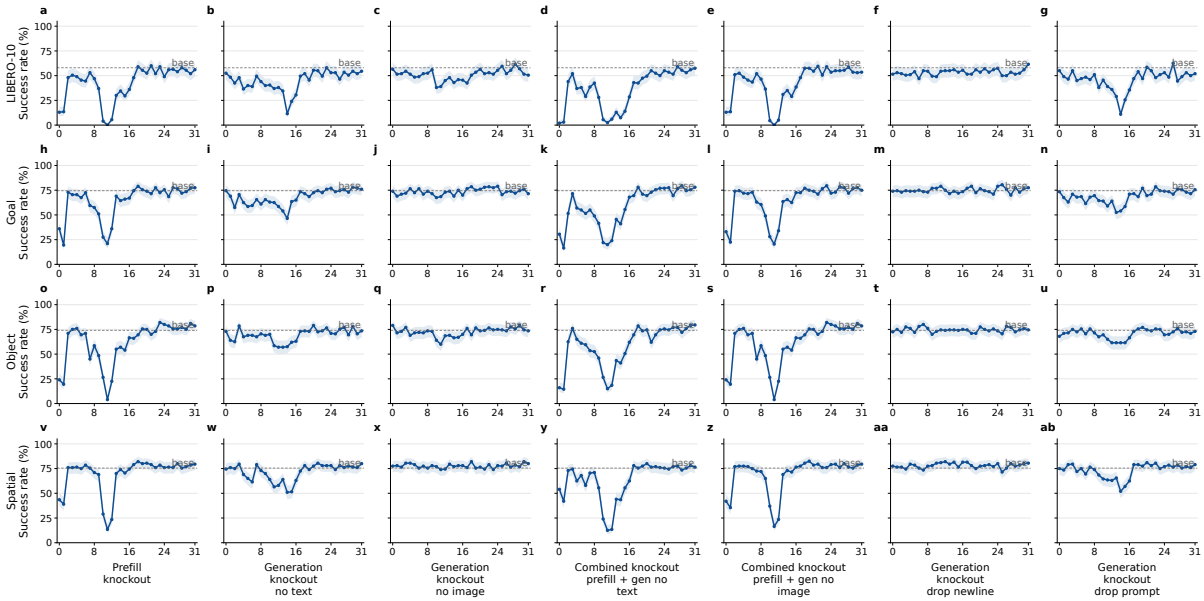

Figure 9: Layer-wise knockout results for **OpenVLA** on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **3-layer** knockout window centered at the indicated layer.

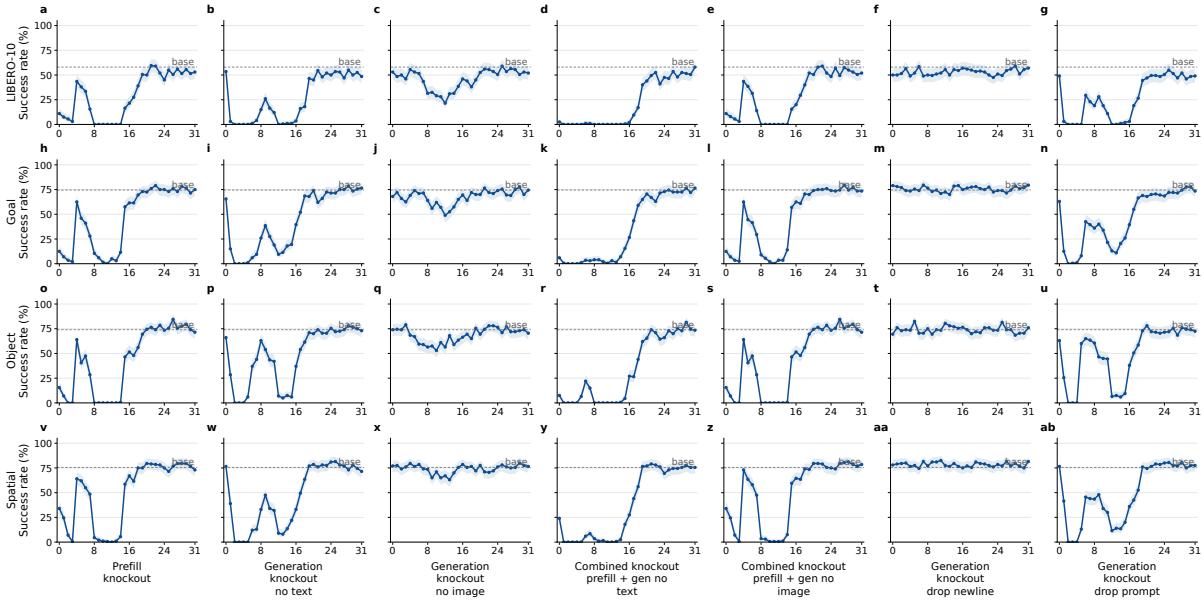

Figure 10: Layer-wise knockout results for **OpenVLA** on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **7-layer** knockout window centered at the indicated layer.

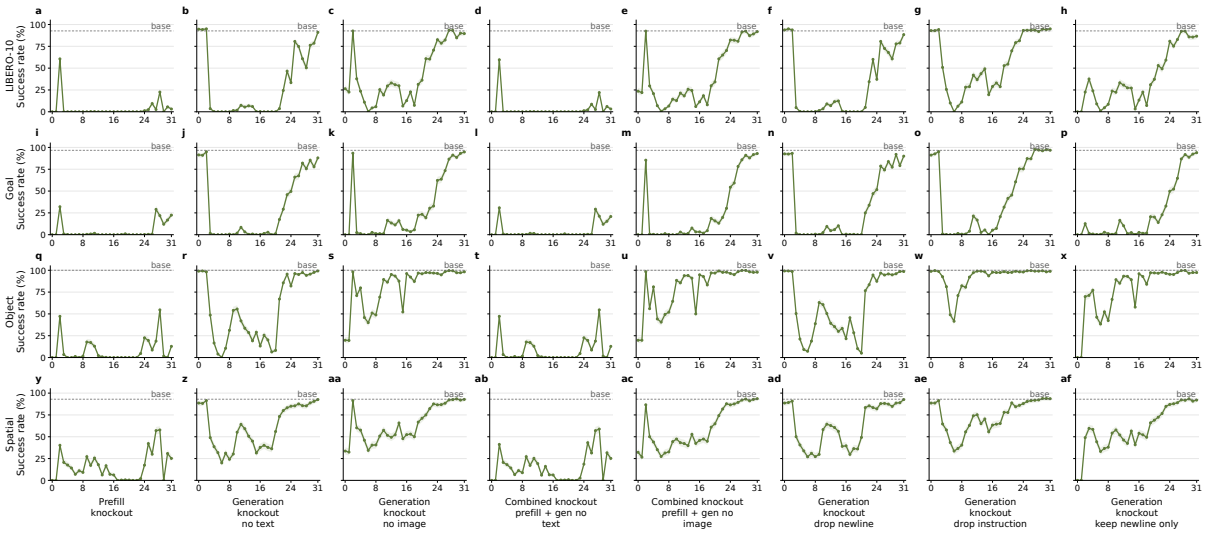

Figure 11: Layer-wise knockout results for **OpenVLA-OFT** on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **3-layer** knockout window centered at the indicated layer.

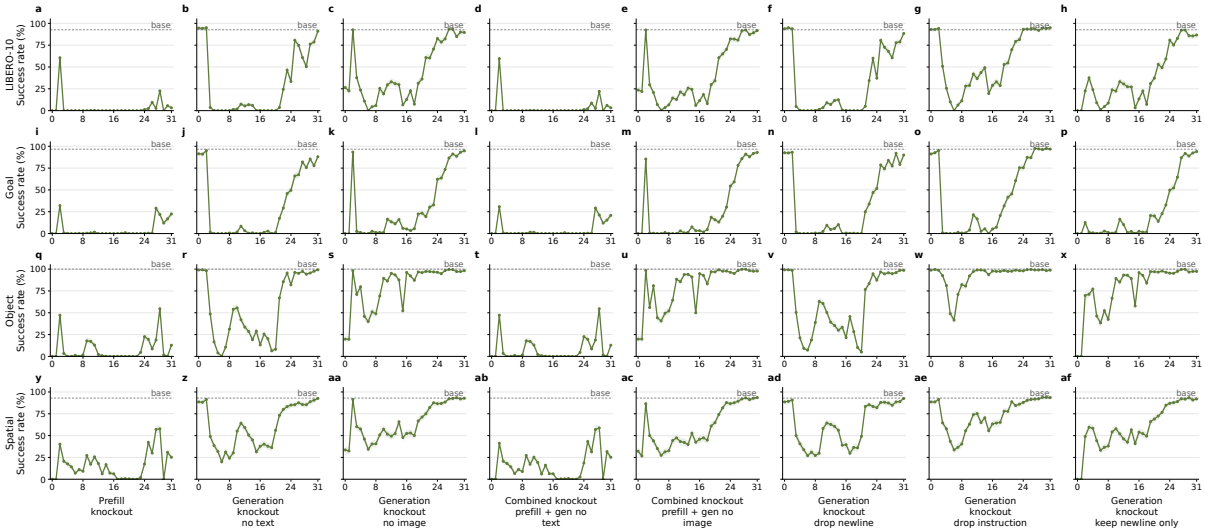

Figure 12: Layer-wise knockout results for **OpenVLA-OFT** on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **5-layer** knockout window centered at the indicated layer.

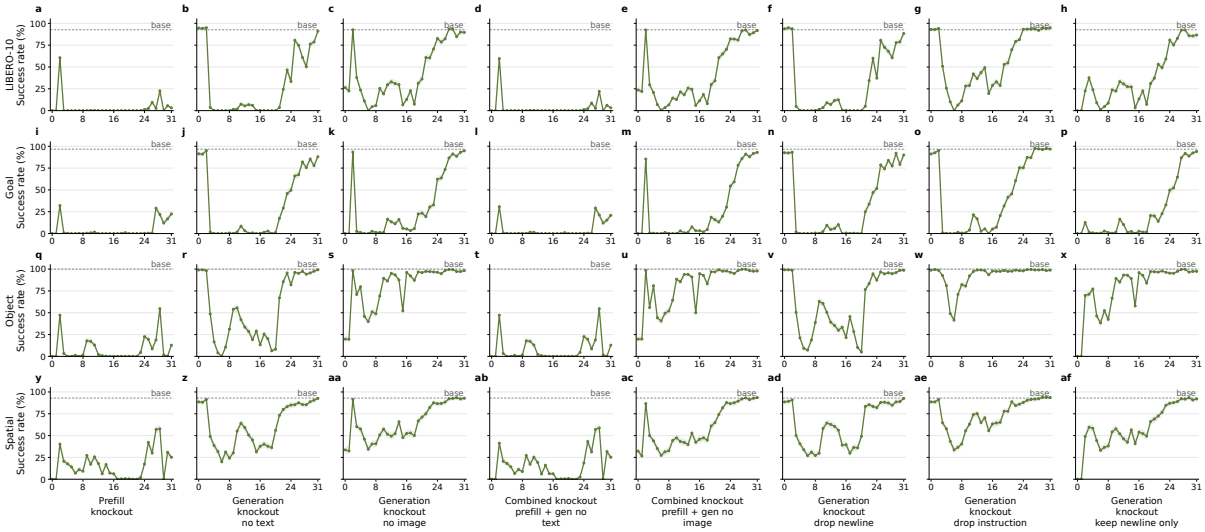

Figure 13: Layer-wise knockout results for **OpenVLA-OFT** on LIBERO-10, Goal, Object, and Spatial. Each point reports the success rate under a **7-layer** knockout window centered at the indicated layer.

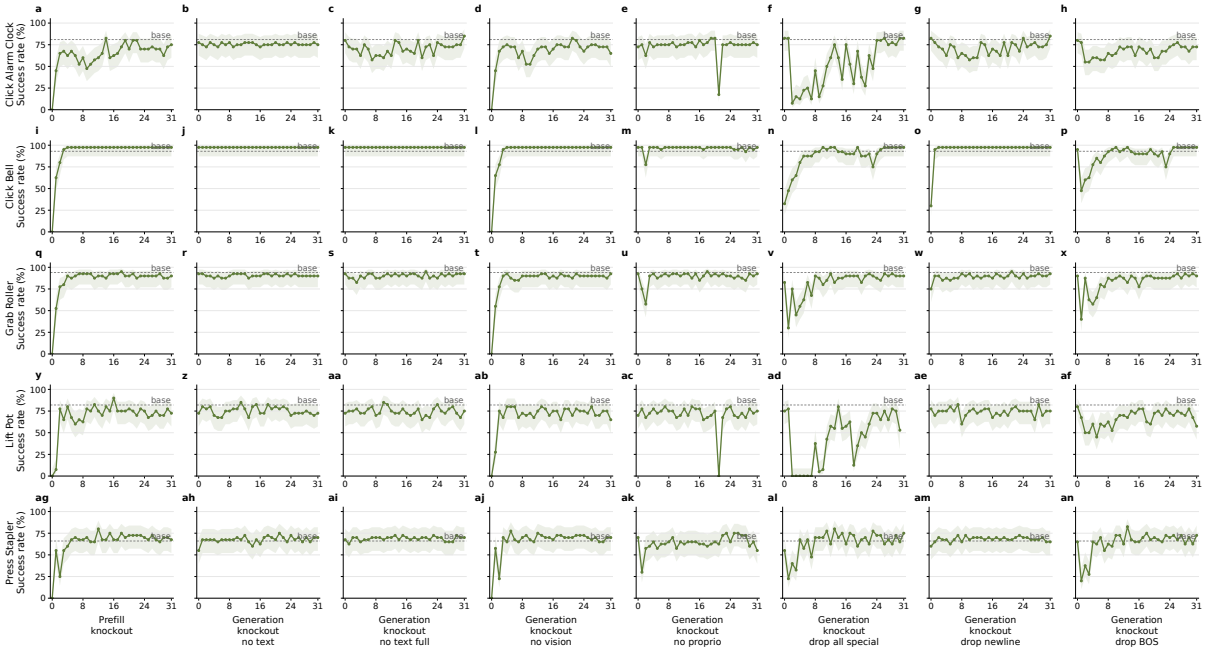

Figure 14: Layer-wise knockout results for **OpenVLA-OFT** on RoboTwin tasks. Each row corresponds to one RoboTwin task, and each point reports the success rate under a **1-layer** knockout window at the indicated layer.

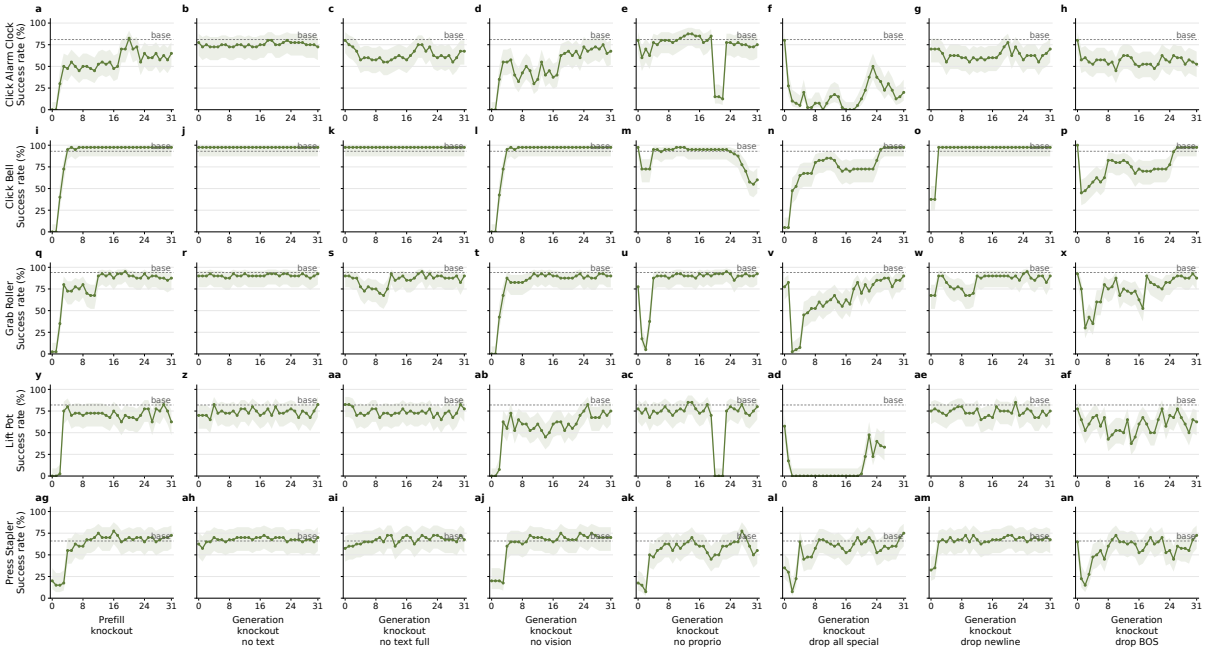

Figure 15: Layer-wise knockout results for **OpenVLA-OFT** on RoboTwin tasks. Each row corresponds to one RoboTwin task, and each point reports the success rate under a **3-layer** knockout window at the indicated layer.

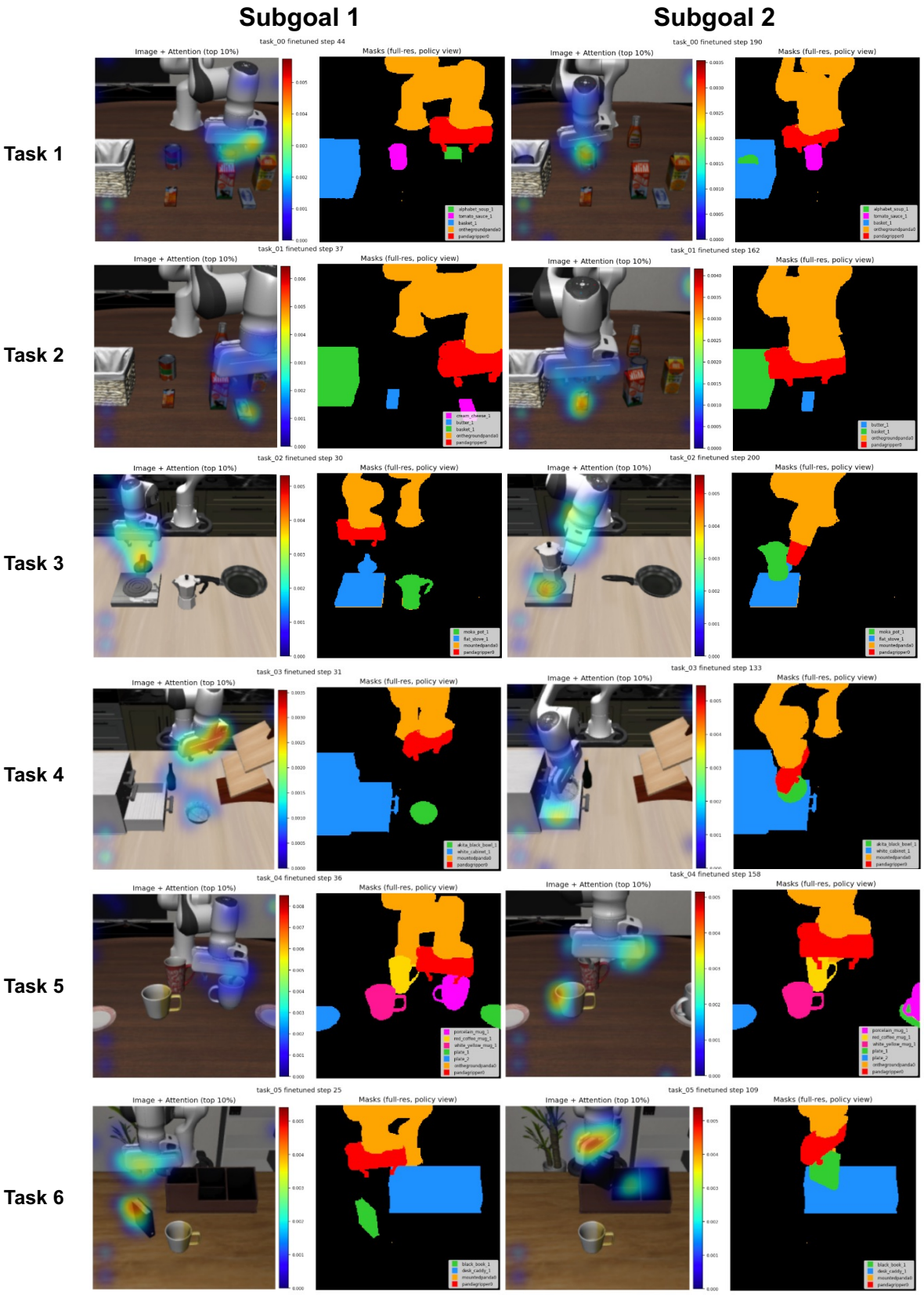

Figure 16: Schematic illustration of attention-IoU computation. We project action-conditioned attention onto the image patch grid, keep the top 10% highest-attention patches, and compute IoU against a simulator-derived target mask. The same mask is also used to compute continuous attention mass and peak-hit rate.

Figure 17: Qualitative action-to-image attention visualizations across rollout stages. We compare pretrained and fine-tuned $\pi_{0.5}$ and OpenVLA models across early, middle, and final execution stages. Finetuning shifts action attention from diffuse or background-biased regions toward task-relevant robot-object interaction regions, indicating stronger visually grounded action generation during manipulation.

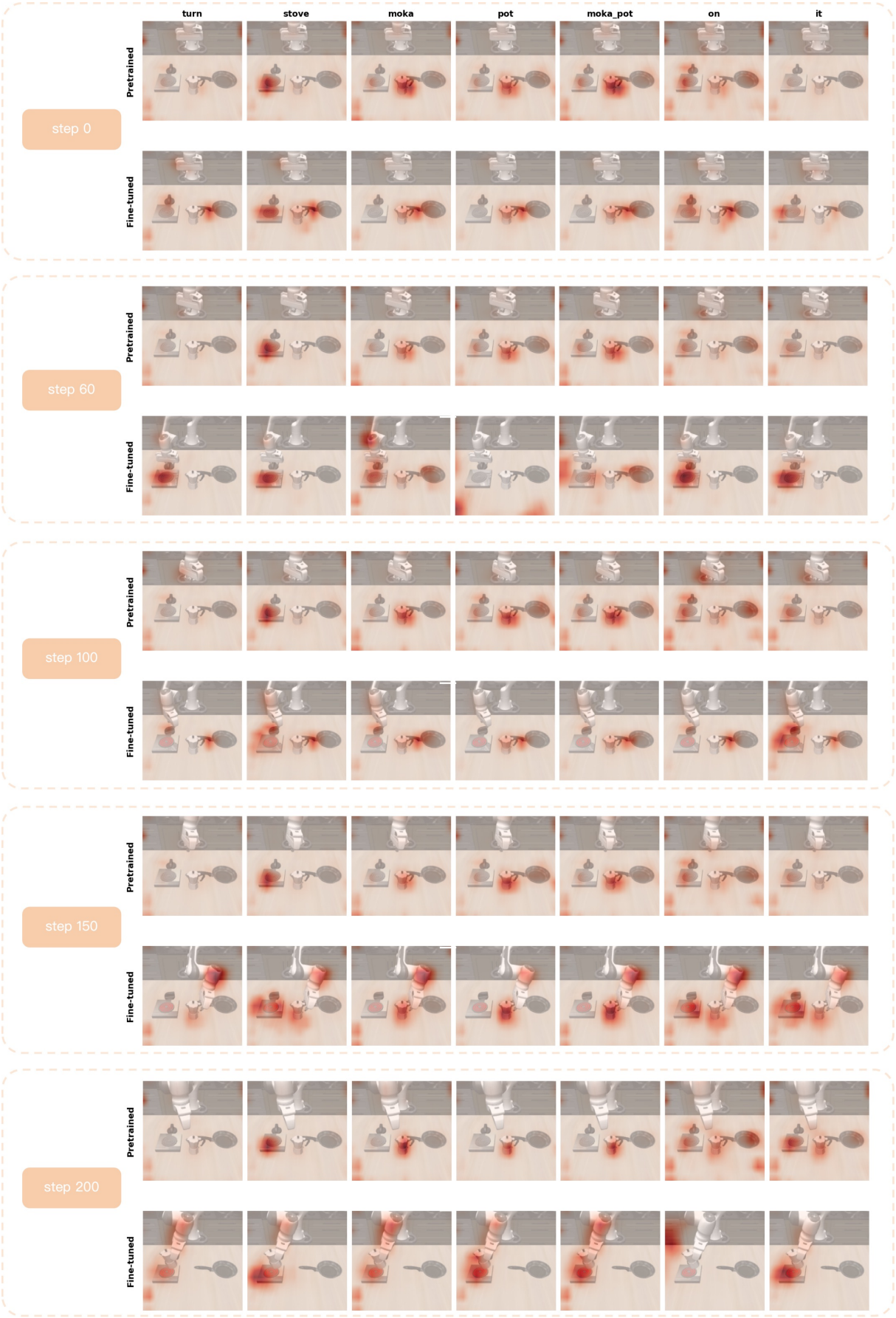

Figure 18: Token-wise text-to-image attention for pretrained and fine-tuned $\pi_{0.5}$ across execution steps. Columns correspond to instruction tokens, and rows compare checkpoints. Each heatmap shows how the selected text token attends to image patches.

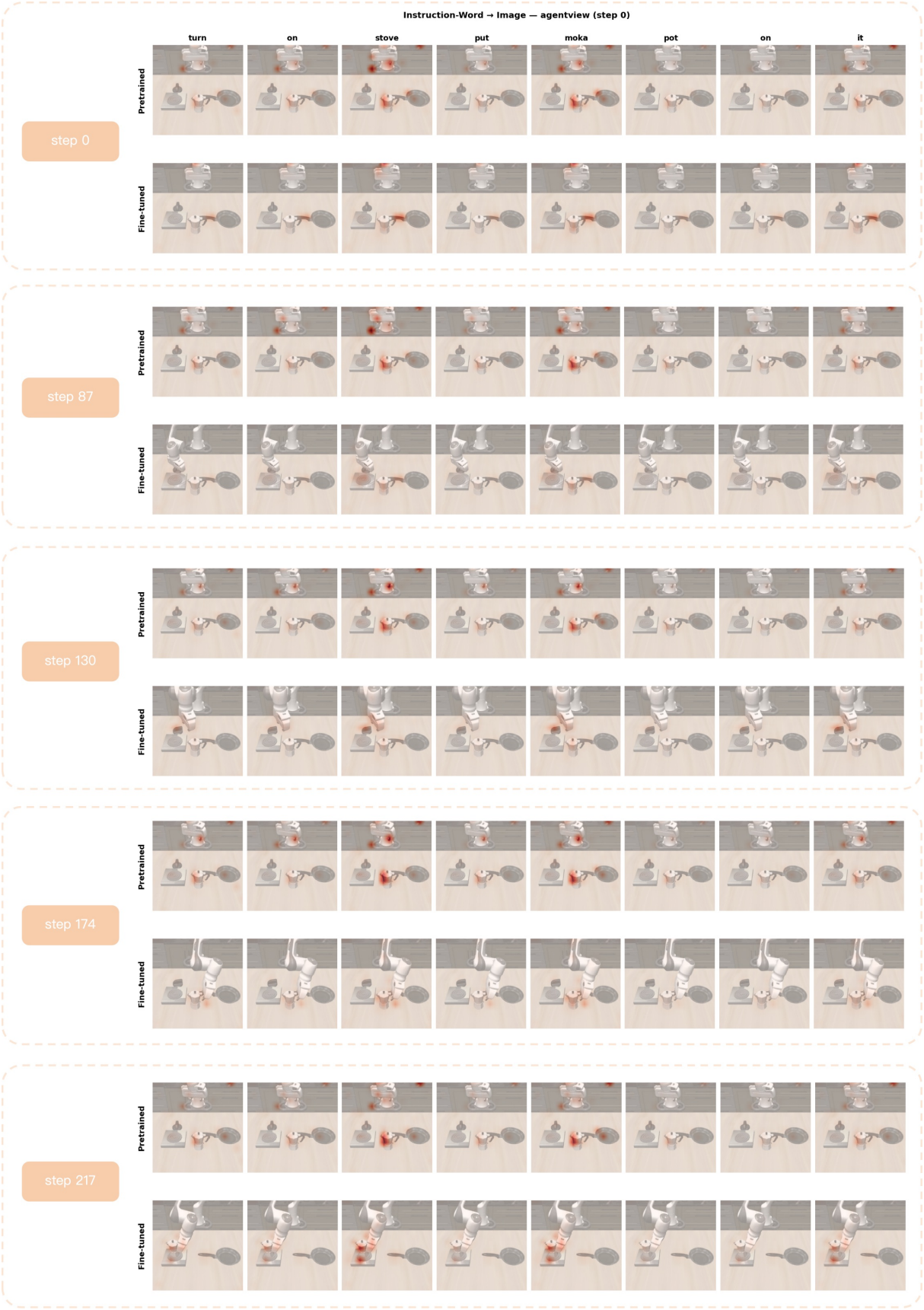

Figure 19: Token-wise text-to-image attention for pretrained and fine-tuned OpenVLA across execution steps. Columns correspond to instruction tokens, and rows compare checkpoints. Each heatmap shows how the selected text token attends to image patches.

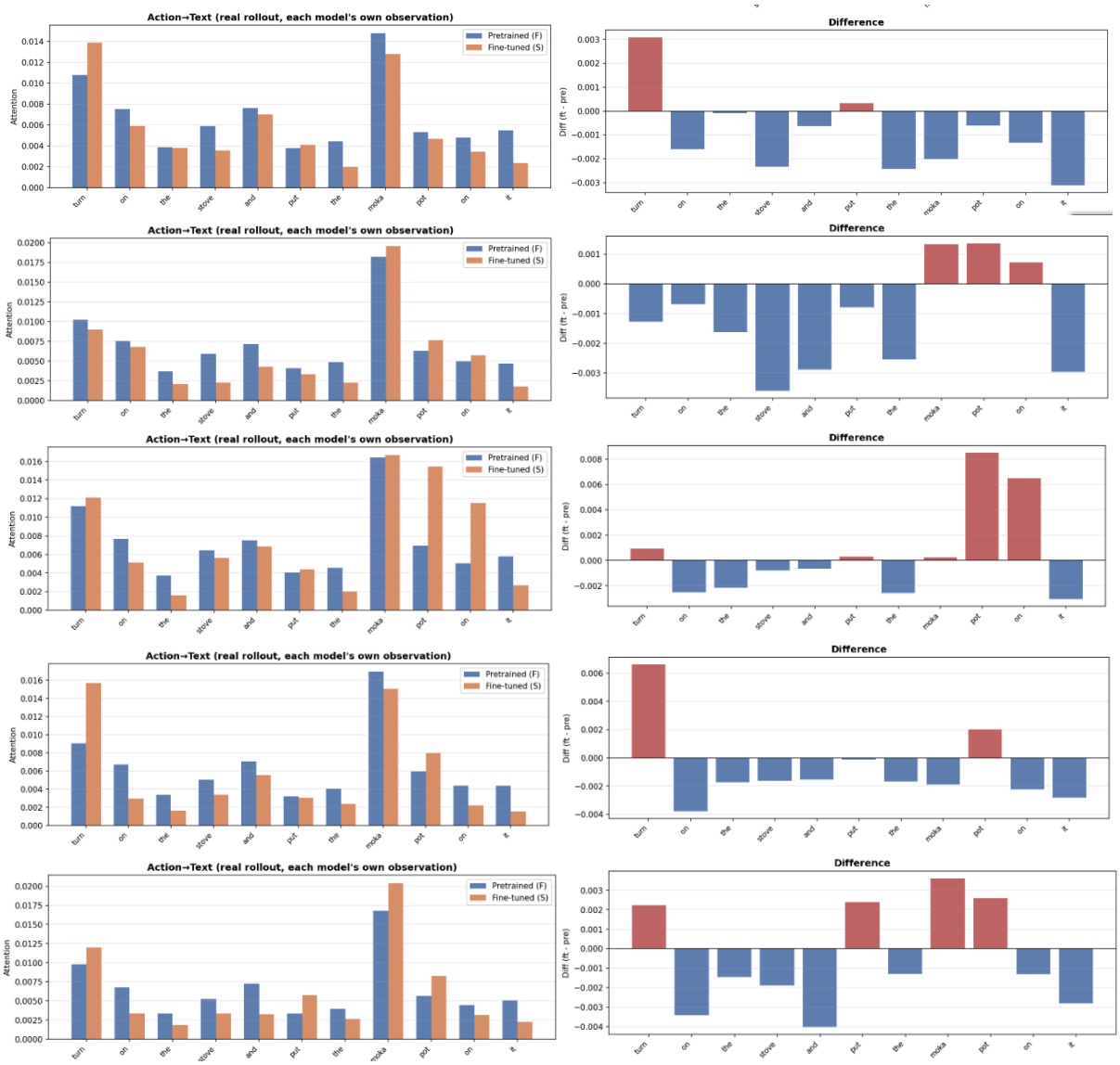

Figure 20: Visualization of action-to-text attention at different timesteps (top to bottom: steps 0, 30, 60, 100, and 250).

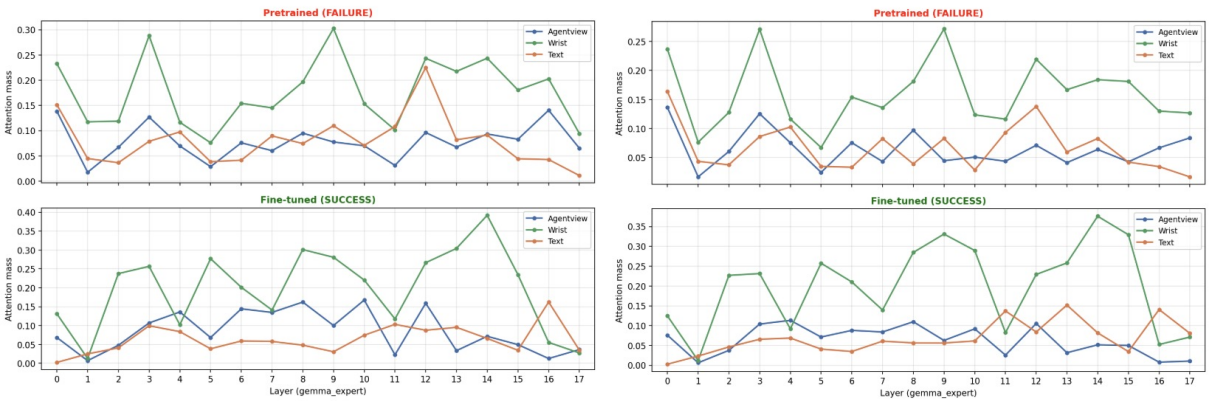

Figure 21: Visualization of layer-wise modality attention (left: step 30, right: step 150) (Top: pretrained, bottom: fine-tuned).

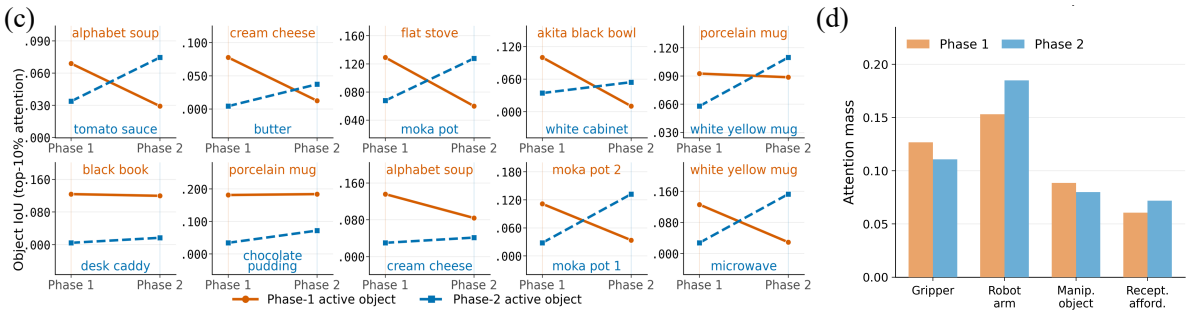

Figure 22: OpenVLA attention IoU and mass on LIBERO-10. (a) Object IoU dynamically shifts between the first (Phase 1) and second (Phase 2) instruction subgoals. (b) Attention mass allocation over robot and object regions. These results indicate that VLA policies successfully generate visually grounded trajectories by tracking task-relevant objects over time.

Model	Setting	Avg	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9
$\pi_{0.5}$	Baseline	75.00	56.00	98.00	64.00	90.00	76.00	86.00	70.00	96.00	58.00	56.00
	Target (BG)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Target (Black)	19.60	12.00	64.00	20.00	2.00	0.00	60.00	0.00	38.00	0.00	0.00
	Target (Mosaic)	47.20	82.00	96.00	34.00	8.00	22.00	94.00	22.00	58.00	54.00	2.00
	Gripper (BG)	8.40	6.00	0.00	0.00	4.00	18.00	54.00	0.00	2.00	0.00	0.00
	Gripper (Black)	52.60	60.00	96.00	6.00	60.00	46.00	68.00	52.00	78.00	24.00	36.00
	Gripper (Mosaic)	53.20	62.00	90.00	2.00	56.00	72.00	84.00	82.00	60.00	14.00	10.00
	Robot (BG)	3.40	0.00	0.00	0.00	2.00	12.00	20.00	0.00	0.00	0.00	0.00
	Robot (Black)	30.20	50.00	90.00	0.00	32.00	8.00	34.00	34.00	34.00	0.00	20.00
	Robot (Mosaic)	39.20	54.00	64.00	0.00	32.00	64.00	38.00	72.00	66.00	2.00	0.00
	Robot w/o Gripper (BG)	52.60	70.00	96.00	0.00	36.00	66.00	44.00	44.00	94.00	44.00	32.00
	Robot w/o Gripper (Black)	58.40	80.00	94.00	0.00	84.00	58.00	58.00	52.00	90.00	26.00	42.00
	Robot w/o Gripper (Mosaic)	64.40	66.00	98.00	4.00	92.00	72.00	62.00	78.00	86.00	58.00	28.00
	Background (Black)	41.00	56.00	70.00	0.00	14.00	52.00	84.00	52.00	76.00	0.00	6.00
	Background (Mosaic)	57.80	60.00	98.00	8.00	94.00	56.00	88.00	54.00	44.00	42.00	34.00
OpenVLA	Baseline	54.33	50.00	83.33	56.67	36.67	50.00	70.00	36.67	76.67	43.33	40.00
	Target (BG)	5.00	0.00	0.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00
	Target (Black)	20.67	33.33	43.33	10.00	0.00	26.67	73.33	3.33	16.67	0.00	0.00
	Target (Mosaic)	25.00	16.67	60.00	6.67	3.33	16.67	76.67	10.00	50.00	6.67	3.33
	Gripper (BG)	10.67	10.00	6.67	23.33	10.00	10.00	3.33	40.00	3.33	0.00	0.00
	Gripper (Black)	31.33	23.33	46.67	36.67	16.67	30.00	36.67	30.00	53.33	3.33	36.67
	Gripper (Mosaic)	20.33	33.33	23.33	50.00	6.67	20.00	30.00	13.33	20.00	6.67	0.00
	Robot (BG)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot (Black)	0.67	0.00	0.00	0.00	0.00	0.00	6.67	0.00	0.00	0.00	0.00
	Robot (Mosaic)	0.33	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot w/o Gripper (BG)	10.00	10.00	13.33	0.00	0.00	26.67	43.33	6.67	0.00	0.00	0.00
	Robot w/o Gripper (Black)	11.67	13.33	16.67	0.00	0.00	23.33	46.67	3.33	10.00	3.33	0.00
	Robot w/o Gripper (Mosaic)	16.00	30.00	23.33	0.00	0.00	16.67	36.67	23.33	30.00	0.00	0.00
	Background (Black)	26.33	30.00	46.67	10.00	16.67	20.00	53.33	10.00	36.67	6.67	33.33
	Background (Mosaic)	48.00	46.67	73.33	30.00	36.67	56.67	76.67	56.67	53.33	20.00	30.00
OpenVLA-OFT	Baseline	94.80	98.00	96.00	100.00	94.00	98.00	100.00	82.00	100.00	88.00	92.00
	Target (BG)	10.40	0.00	0.00	0.00	0.00	0.00	100.00	0.00	4.00	0.00	0.00
	Target (Black)	15.20	10.00	16.00	2.00	0.00	0.00	100.00	0.00	24.00	0.00	0.00
	Target (Mosaic)	48.20	86.00	98.00	0.00	78.00	14.00	100.00	4.00	96.00	0.00	6.00
	Gripper (BG)	89.80	92.00	100.00	92.00	94.00	100.00	98.00	86.00	86.00	64.00	86.00
	Gripper (Black)	89.80	92.00	98.00	98.00	90.00	96.00	94.00	84.00	98.00	60.00	88.00
	Gripper (Mosaic)	95.00	98.00	94.00	92.00	98.00	98.00	96.00	92.00	98.00	88.00	96.00
	Robot (BG)	66.80	68.00	66.00	54.00	82.00	88.00	80.00	50.00	88.00	30.00	62.00
	Robot (Black)	79.00	86.00	92.00	66.00	86.00	96.00	84.00	52.00	94.00	52.00	82.00
	Robot (Mosaic)	84.40	90.00	96.00	70.00	100.00	92.00	88.00	92.00	98.00	30.00	88.00
	Robot w/o Gripper (BG)	85.40	88.00	88.00	94.00	90.00	96.00	98.00	54.00	100.00	58.00	88.00
	Robot w/o Gripper (Black)	84.00	92.00	94.00	96.00	84.00	98.00	96.00	46.00	98.00	56.00	80.00
	Robot w/o Gripper (Mosaic)	91.20	92.00	96.00	94.00	100.00	98.00	98.00	94.00	98.00	50.00	92.00
	Background (Black)	73.80	100.00	92.00	86.00	90.00	92.00	84.00	62.00	96.00	8.00	28.00
	Background (Mosaic)	84.60	92.00	98.00	80.00	92.00	84.00	100.00	76.00	100.00	40.00	84.00
X-VLA	Baseline	96.00	92.00	98.00	100.00	92.00	96.00	96.00	94.00	98.00	96.00	98.00
	Target (BG)	46.40	70.00	46.00	86.00	26.00	22.00	80.00	38.00	28.00	40.00	28.00
	Target (Black)	78.60	82.00	98.00	92.00	72.00	26.00	94.00	72.00	86.00	100.00	64.00
	Target (Mosaic)	91.40	94.00	100.00	92.00	96.00	76.00	84.00	94.00	90.00	90.00	98.00
	Gripper (BG)	90.80	80.00	90.00	96.00	88.00	96.00	92.00	86.00	96.00	94.00	90.00
	Gripper (Black)	96.60	98.00	100.00	100.00	92.00	92.00	92.00	98.00	96.00	100.00	98.00
	Gripper (Mosaic)	95.60	94.00	98.00	100.00	88.00	98.00	90.00	94.00	96.00	98.00	100.00
	Robot (BG)	43.33	60.00	66.67	23.33	16.67	36.67	53.33	46.67	76.67	6.67	46.67
	Robot (Black)	83.40	78.00	100.00	80.00	80.00	92.00	84.00	96.00	90.00	68.00	66.00
	Robot (Mosaic)	96.20	96.00	100.00	100.00	96.00	100.00	86.00	92.00	96.00	100.00	96.00
	Robot w/o Gripper (BG)	80.40	90.00	100.00	92.00	82.00	100.00	84.00	92.00	98.00	14.00	52.00
	Robot w/o Gripper (Black)	88.20	92.00	100.00	92.00	80.00	96.00	84.00	94.00	100.00	86.00	58.00
	Robot w/o Gripper (Mosaic)	96.80	98.00	100.00	100.00	100.00	98.00	88.00	92.00	100.00	94.00	98.00
	Background (Black)	89.00	94.00	92.00	96.00	84.00	96.00	36.00	94.00	100.00	100.00	98.00
	Background (Mosaic)	89.40	94.00	98.00	98.00	92.00	94.00	34.00	92.00	98.00	96.00	98.00

Table 7: Success rates (%) across tasks and masking strategies on **LIBERO-10** for $\pi_{0.5}$, OpenVLA, OpenVLA-OFT and X-VLA. The instruction is provided in Tab. 15.

Model	Setting	Avg	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9
$\pi_{0.5}$	Baseline	95.60	96.00	98.00	78.00	98.00	90.00	100.00	100.00	96.00	100.00	100.00
	Target (BG)	13.00	8.00	0.00	0.00	0.00	0.00	0.00	52.00	0.00	70.00	0.00
	Target (Black)	61.60	16.00	94.00	44.00	94.00	94.00	24.00	72.00	90.00	88.00	0.00
	Target (Mosaic)	74.00	58.00	94.00	22.00	86.00	96.00	14.00	90.00	90.00	92.00	98.00
	Gripper (BG)	3.20	2.00	2.00	0.00	0.00	24.00	0.00	0.00	0.00	4.00	0.00
	Gripper (Black)	97.40	100.00	100.00	86.00	96.00	100.00	98.00	98.00	96.00	100.00	100.00
	Gripper (Mosaic)	79.40	100.00	96.00	24.00	76.00	100.00	26.00	100.00	100.00	98.00	74.00
	Robot (BG)	26.40	8.00	52.00	8.00	24.00	64.00	0.00	10.00	58.00	40.00	0.00
	Robot (Black)	93.60	94.00	98.00	90.00	84.00	92.00	98.00	90.00	100.00	96.00	94.00
	Robot (Mosaic)	74.60	100.00	96.00	26.00	76.00	100.00	20.00	100.00	100.00	94.00	34.00
	Robot w/o Gripper (BG)	91.80	94.00	98.00	90.00	76.00	66.00	98.00	100.00	96.00	100.00	100.00
	Robot w/o Gripper (Black)	92.80	94.00	100.00	90.00	88.00	66.00	98.00	96.00	100.00	100.00	96.00
	Robot w/o Gripper (Mosaic)	96.60	100.00	98.00	80.00	98.00	96.00	100.00	98.00	98.00	100.00	98.00
	Background (Black)	79.40	66.00	100.00	80.00	78.00	100.00	68.00	62.00	68.00	88.00	84.00
	Background (Mosaic)	86.60	96.00	96.00	22.00	78.00	100.00	88.00	92.00	98.00	100.00	96.00
OpenVLA	Baseline	70.00	63.33	76.67	70.00	46.67	83.33	73.33	63.33	76.67	60.00	86.67
	Target (BG)	0.33	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00	0.00	0.00
	Target (Black)	55.67	83.33	46.67	73.33	43.33	63.33	80.00	50.00	73.33	20.00	23.33
	Target (Mosaic)	59.67	70.00	50.00	56.67	26.67	76.67	63.33	63.33	86.67	60.00	43.33
	Gripper (BG)	1.00	0.00	0.00	6.67	3.33	0.00	0.00	0.00	0.00	0.00	0.00
	Gripper (Black)	72.00	73.33	40.00	80.00	56.67	80.00	80.00	86.67	96.67	43.33	83.33
	Gripper (Mosaic)	3.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	36.67
	Robot (BG)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot (Black)	4.00	0.00	0.00	0.00	0.00	0.00	36.67	0.00	0.00	0.00	3.33
	Robot (Mosaic)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot w/o Gripper (BG)	34.33	73.33	33.33	43.33	56.67	33.33	6.67	0.00	6.67	10.00	80.00
	Robot w/o Gripper (Black)	17.33	0.00	16.67	36.67	26.67	0.00	23.33	0.00	0.00	0.00	70.00
	Robot w/o Gripper (Mosaic)	15.67	50.00	13.33	10.00	30.00	3.33	6.67	0.00	6.67	0.00	36.67
	Background (Black)	42.67	63.33	23.33	43.33	40.00	66.67	0.00	6.67	73.33	50.00	60.00
	Background (Mosaic)	65.67	76.67	56.67	83.33	53.33	93.33	63.33	40.00	76.67	40.00	73.33
OpenVLA-OFT	Baseline	99.80	100.00	100.00	98.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Target (BG)	57.00	94.00	0.00	16.00	46.00	38.00	80.00	94.00	12.00	96.00	94.00
	Target (Black)	84.80	70.00	32.00	100.00	96.00	100.00	100.00	98.00	80.00	74.00	98.00
	Target (Mosaic)	98.40	94.00	96.00	98.00	98.00	100.00	98.00	100.00	100.00	100.00	100.00
	Gripper (BG)	94.20	100.00	100.00	100.00	94.00	66.00	100.00	96.00	88.00	100.00	98.00
	Gripper (Black)	98.80	96.00	100.00	100.00	98.00	100.00	100.00	96.00	100.00	98.00	100.00
	Gripper (Mosaic)	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Robot (BG)	96.00	98.00	100.00	100.00	98.00	92.00	100.00	92.00	86.00	96.00	98.00
	Robot (Black)	96.80	98.00	98.00	96.00	98.00	96.00	98.00	92.00	96.00	100.00	96.00
	Robot (Mosaic)	97.60	100.00	92.00	100.00	98.00	96.00	100.00	96.00	96.00	98.00	100.00
	Robot w/o Gripper (BG)	98.60	100.00	100.00	98.00	100.00	100.00	98.00	98.00	100.00	100.00	92.00
	Robot w/o Gripper (Black)	99.80	100.00	100.00	100.00	100.00	100.00	100.00	98.00	100.00	100.00	100.00
	Robot w/o Gripper (Mosaic)	98.80	100.00	98.00	100.00	96.00	100.00	98.00	98.00	100.00	100.00	98.00
	Background (Black)	47.40	96.00	2.00	2.00	16.00	86.00	52.00	70.00	24.00	100.00	26.00
	Background (Mosaic)	95.00	100.00	86.00	98.00	100.00	96.00	88.00	94.00	98.00	100.00	90.00
X-VLA	Baseline	98.60	100.00	100.00	100.00	96.00	98.00	94.00	100.00	98.00	100.00	100.00
	Target (BG)	94.20	98.00	100.00	64.00	100.00	100.00	88.00	100.00	98.00	98.00	96.00
	Target (Black)	97.40	100.00	100.00	100.00	98.00	100.00	76.00	100.00	100.00	100.00	100.00
	Target (Mosaic)	98.40	100.00	100.00	100.00	94.00	100.00	90.00	100.00	100.00	100.00	100.00
	Gripper (BG)	95.60	98.00	100.00	100.00	98.00	100.00	62.00	100.00	98.00	100.00	100.00
	Gripper (Black)	97.80	100.00	100.00	100.00	96.00	100.00	84.00	100.00	98.00	100.00	100.00
	Gripper (Mosaic)	97.20	100.00	100.00	100.00	100.00	100.00	74.00	100.00	98.00	100.00	100.00
	Robot (BG)	63.20	96.00	42.00	96.00	82.00	62.00	30.00	42.00	78.00	22.00	82.00
	Robot (Black)	97.80	100.00	100.00	100.00	96.00	100.00	82.00	100.00	100.00	100.00	100.00
	Robot (Mosaic)	97.40	100.00	100.00	100.00	100.00	98.00	78.00	100.00	98.00	100.00	100.00
	Robot w/o Gripper (BG)	93.20	100.00	92.00	98.00	92.00	100.00	90.00	80.00	98.00	88.00	94.00
	Robot w/o Gripper (Black)	98.40	100.00	100.00	98.00	94.00	100.00	96.00	100.00	98.00	100.00	98.00
	Robot w/o Gripper (Mosaic)	98.40	100.00	100.00	100.00	96.00	100.00	90.00	100.00	98.00	100.00	100.00
	Background (Black)	91.00	100.00	82.00	40.00	92.00	100.00	98.00	100.00	98.00	100.00	100.00
	Background (Mosaic)	98.40	98.00	100.00	98.00	94.00	100.00	96.00	100.00	98.00	100.00	100.00

Table 8: Success rates (%) across tasks and masking strategies on **LIBERO-Object** for $\pi_{0.5}$, OpenVLA, OpenVLA-OFT, and X-VLA. The instruction is provided in Tab. 15.

Model	Setting	Avg	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9
$\pi_{0.5}$	Baseline	95.80	100.00	92.00	100.00	100.00	88.00	100.00	100.00	96.00	86.00	96.00
	Target (BG)	7.80	0.00	0.00	0.00	0.00	8.00	0.00	0.00	52.00	0.00	18.00
	Target (Black)	23.00	44.00	12.00	64.00	6.00	0.00	2.00	24.00	64.00	0.00	14.00
	Target (Mosaic)	50.20	74.00	64.00	46.00	62.00	20.00	16.00	78.00	76.00	20.00	46.00
	Gripper (BG)	64.00	76.00	66.00	100.00	64.00	20.00	72.00	84.00	98.00	60.00	0.00
	Gripper (Black)	94.80	100.00	94.00	100.00	96.00	78.00	98.00	100.00	100.00	90.00	92.00
	Gripper (Mosaic)	89.80	100.00	94.00	100.00	92.00	58.00	94.00	100.00	98.00	90.00	72.00
	Robot (BG)	47.60	78.00	2.00	92.00	44.00	4.00	68.00	94.00	86.00	8.00	0.00
	Robot (Black)	95.00	100.00	98.00	98.00	94.00	84.00	98.00	100.00	96.00	94.00	88.00
	Robot (Mosaic)	89.60	100.00	98.00	100.00	82.00	66.00	94.00	98.00	100.00	88.00	70.00
	Robot w/o Gripper (BG)	97.40	98.00	100.00	98.00	100.00	92.00	96.00	100.00	100.00	92.00	98.00
	Robot w/o Gripper (Black)	99.00	98.00	98.00	100.00	100.00	98.00	100.00	100.00	100.00	96.00	100.00
	Robot w/o Gripper (Mosaic)	96.40	100.00	96.00	96.00	100.00	92.00	96.00	100.00	94.00	92.00	98.00
	Background (Black)	67.20	96.00	46.00	94.00	58.00	54.00	30.00	100.00	92.00	26.00	76.00
	Background (Mosaic)	93.00	100.00	96.00	100.00	96.00	68.00	100.00	98.00	94.00	84.00	94.00
OpenVLA	Baseline	79.67	86.67	90.00	80.00	100.00	76.67	46.67	86.67	76.67	83.33	70.00
	Target (BG)	36.67	26.67	83.33	43.33	36.67	3.33	0.00	66.67	70.00	16.67	20.00
	Target (Black)	63.33	60.00	93.33	63.33	86.67	63.33	3.33	80.00	90.00	50.00	43.33
	Target (Mosaic)	70.00	83.33	93.33	76.67	83.33	60.00	30.00	86.67	80.00	63.33	43.33
	Gripper (BG)	19.00	23.33	0.00	10.00	26.67	13.33	0.00	23.33	40.00	53.33	0.00
	Gripper (Black)	70.67	80.00	70.00	93.33	86.67	56.67	13.33	90.00	80.00	73.33	63.33
	Gripper (Mosaic)	53.00	56.67	73.33	46.67	90.00	6.67	0.00	73.33	60.00	76.67	46.67
	Robot (BG)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot (Black)	3.33	0.00	13.33	0.00	0.00	3.33	0.00	0.00	0.00	16.67	0.00
	Robot (Mosaic)	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.67	3.33
	Robot w/o Gripper (BG)	22.00	13.33	16.67	46.67	43.33	26.67	0.00	10.00	0.00	23.33	40.00
	Robot w/o Gripper (Black)	31.67	46.67	46.67	33.33	70.00	46.67	0.00	33.33	6.67	23.33	10.00
	Robot w/o Gripper (Mosaic)	49.67	43.33	76.67	73.33	60.00	66.67	0.00	33.33	43.33	63.33	36.67
	Background (Black)	47.33	43.33	3.33	46.67	100.00	56.67	3.33	83.33	23.33	70.00	43.33
	Background (Mosaic)	82.33	96.67	96.67	80.00	86.67	66.67	46.67	93.33	90.00	90.00	76.67
OpenVLA-OFT	Baseline	92.80	100.00	98.00	100.00	100.00	96.00	46.00	100.00	94.00	96.00	98.00
	Target (BG)	26.80	26.00	0.00	10.00	26.00	56.00	0.00	30.00	18.00	10.00	92.00
	Target (Black)	35.40	14.00	54.00	0.00	58.00	42.00	2.00	44.00	18.00	32.00	90.00
	Target (Mosaic)	76.00	92.00	90.00	92.00	84.00	90.00	6.00	96.00	38.00	76.00	96.00
	Gripper (BG)	92.40	100.00	100.00	100.00	100.00	100.00	36.00	98.00	96.00	96.00	98.00
	Gripper (Black)	92.80	100.00	100.00	98.00	100.00	100.00	46.00	98.00	96.00	94.00	96.00
	Gripper (Mosaic)	93.80	100.00	98.00	100.00	100.00	98.00	54.00	100.00	92.00	96.00	100.00
	Robot (BG)	81.40	86.00	84.00	100.00	94.00	88.00	30.00	82.00	90.00	66.00	94.00
	Robot (Black)	86.80	88.00	94.00	98.00	98.00	98.00	30.00	100.00	94.00	74.00	94.00
	Robot (Mosaic)	91.60	98.00	100.00	100.00	100.00	90.00	38.00	100.00	94.00	96.00	100.00
	Robot w/o Gripper (BG)	90.40	90.00	100.00	100.00	96.00	94.00	48.00	100.00	92.00	90.00	94.00
	Robot w/o Gripper (Black)	91.00	96.00	100.00	100.00	100.00	98.00	40.00	100.00	92.00	90.00	94.00
	Robot w/o Gripper (Mosaic)	91.20	98.00	100.00	96.00	100.00	98.00	34.00	100.00	96.00	94.00	96.00
	Background (Black)	83.20	96.00	88.00	100.00	100.00	82.00	0.00	100.00	94.00	86.00	86.00
	Background (Mosaic)	88.20	100.00	86.00	100.00	98.00	92.00	24.00	100.00	94.00	94.00	94.00
X-VLA	Baseline	98.00	98.00	100.00	100.00	100.00	90.00	98.00	100.00	94.00	100.00	100.00
	Target (BG)	79.80	66.00	84.00	64.00	92.00	84.00	86.00	90.00	58.00	82.00	92.00
	Target (Black)	90.20	56.00	94.00	100.00	100.00	90.00	98.00	98.00	78.00	94.00	94.00
	Target (Mosaic)	89.80	92.00	90.00	74.00	100.00	96.00	88.00	100.00	74.00	94.00	90.00
	Gripper (BG)	97.00	92.00	100.00	98.00	100.00	94.00	98.00	100.00	94.00	100.00	94.00
	Gripper (Black)	97.20	100.00	100.00	100.00	98.00	90.00	96.00	100.00	94.00	98.00	96.00
	Gripper (Mosaic)	96.80	100.00	100.00	100.00	100.00	90.00	96.00	98.00	92.00	98.00	94.00
	Robot (BG)	80.80	86.00	88.00	94.00	98.00	88.00	12.00	80.00	72.00	100.00	90.00
	Robot (Black)	96.40	100.00	100.00	100.00	100.00	96.00	92.00	98.00	88.00	98.00	92.00
	Robot (Mosaic)	96.20	100.00	98.00	100.00	100.00	88.00	96.00	98.00	96.00	94.00	92.00
	Robot w/o Gripper (BG)	98.00	100.00	100.00	100.00	100.00	94.00	100.00	92.00	96.00	98.00	100.00
	Robot w/o Gripper (Black)	97.60	100.00	100.00	100.00	100.00	90.00	96.00	100.00	90.00	100.00	100.00
	Robot w/o Gripper (Mosaic)	97.20	100.00	100.00	100.00	98.00	88.00	100.00	100.00	92.00	98.00	96.00
	Background (Black)	97.60	100.00	98.00	100.00	98.00	94.00	98.00	100.00	92.00	100.00	96.00
	Background (Mosaic)	97.60	100.00	100.00	100.00	100.00	84.00	98.00	100.00	94.00	100.00	100.00

Table 9: Success rates (%) across tasks and masking strategies on **LIBERO-Spatial** for $\pi_{0.5}$, OpenVLA, OpenVLA-OFT, and X-VLA. The instruction is provided in Tab. 15.

Model	Setting	Avg	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9
$\pi_{0.5}$	Baseline	80.00	8.00	100.00	94.00	46.00	100.00	100.00	100.00	100.00	100.00	52.00
	Target(BG)	19.80	6.00	12.00	52.00	2.00	10.00	6.00	14.00	68.00	20.00	8.00
	Target(Black)	58.80	4.00	70.00	92.00	2.00	76.00	54.00	54.00	100.00	56.00	80.00
	Target(Mosaic)	58.40	6.00	32.00	92.00	8.00	12.00	98.00	92.00	100.00	94.00	50.00
	Gripper(BG)	59.20	0.00	96.00	68.00	6.00	96.00	44.00	90.00	100.00	88.00	4.00
	Gripper(Black)	74.80	0.00	100.00	90.00	28.00	100.00	100.00	100.00	100.00	100.00	30.00
	Gripper(Mosaic)	74.20	2.00	100.00	92.00	16.00	100.00	100.00	100.00	100.00	100.00	32.00
	Robot(BG)	44.00	0.00	92.00	10.00	0.00	94.00	34.00	58.00	52.00	100.00	0.00
	Robot(Black)	69.80	6.00	100.00	78.00	18.00	100.00	98.00	98.00	100.00	100.00	0.00
	Robot(Mosaic)	68.60	6.00	100.00	40.00	18.00	100.00	92.00	98.00	100.00	100.00	32.00
	Robot w/o Gripper(BG)	76.80	2.00	100.00	76.00	46.00	100.00	100.00	100.00	100.00	100.00	44.00
	Robot w/o Gripper(Black)	76.40	8.00	100.00	90.00	32.00	100.00	100.00	100.00	100.00	100.00	34.00
	Robot w/o Gripper(Mosaic)	79.00	18.00	100.00	76.00	44.00	100.00	100.00	96.00	100.00	100.00	56.00
	Background(Black)	57.80	16.00	100.00	20.00	16.00	100.00	40.00	58.00	100.00	100.00	28.00
	Background(Mosaic)	78.00	2.00	100.00	90.00	36.00	100.00	100.00	96.00	100.00	100.00	56.00
OpenVLA	Baseline	74.00	53.33	93.33	80.00	43.33	93.33	80.00	63.33	93.33	83.33	56.67
	Target(BG)	45.00	53.33	60.00	30.00	46.67	73.33	73.33	3.33	46.67	26.67	36.67
	Target(Black)	73.00	73.33	83.33	83.33	46.67	83.33	73.33	50.00	93.33	76.67	66.67
	Target(Mosaic)	67.67	50.00	80.00	73.33	63.33	80.00	76.67	43.33	100.00	66.67	43.33
	Gripper(BG)	38.00	33.33	26.67	33.33	3.33	86.67	63.33	50.00	20.00	43.33	20.00
	Gripper(Black)	69.33	50.00	73.33	83.33	13.33	90.00	73.33	80.00	86.67	86.67	56.67
	Gripper(Mosaic)	63.00	20.00	60.00	63.33	30.00	100.00	70.00	80.00	96.67	63.33	46.67
	Robot(BG)	0.33	0.00	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Robot(Black)	6.00	0.00	3.33	0.00	0.00	3.33	0.00	16.67	36.67	0.00	0.00
	Robot(Mosaic)	8.00	0.00	0.00	3.33	0.00	40.00	10.00	6.67	20.00	0.00	0.00
	Robot w/o Gripper(BG)	25.67	10.00	63.33	13.33	0.00	50.00	0.00	23.33	66.67	30.00	0.00
	Robot w/o Gripper(Black)	26.00	13.33	70.00	30.00	0.00	26.67	10.00	16.67	76.67	16.67	0.00
	Robot w/o Gripper(Mosaic)	38.33	6.67	70.00	53.33	10.00	80.00	30.00	13.33	66.67	53.33	0.00
	Background(Black)	41.67	33.33	56.67	3.33	13.33	63.33	86.67	43.33	80.00	26.67	10.00
	Background(Mosaic)	70.67	56.67	100.00	63.33	30.00	96.67	86.67	50.00	96.67	76.67	50.00
OpenVLA-OFT	Baseline	97.40	98.00	100.00	94.00	96.00	100.00	96.00	94.00	100.00	100.00	96.00
	Target (BG)	49.60	98.00	82.00	68.00	38.00	72.00	0.00	4.00	34.00	18.00	82.00
	Target (Black)	60.20	98.00	90.00	94.00	46.00	86.00	0.00	2.00	86.00	0.00	100.00
	Target (Mosaic)	88.40	98.00	96.00	96.00	82.00	98.00	68.00	74.00	78.00	96.00	98.00
	Gripper (BG)	97.20	100.00	100.00	90.00	90.00	100.00	96.00	98.00	100.00	100.00	98.00
	Gripper (Black)	97.00	98.00	98.00	98.00	90.00	98.00	100.00	96.00	98.00	96.00	98.00
	Gripper (Mosaic)	97.60	98.00	100.00	98.00	90.00	100.00	98.00	94.00	100.00	100.00	98.00
	Robot (BG)	67.20	94.00	50.00	96.00	16.00	100.00	70.00	60.00	18.00	92.00	76.00
	Robot (Black)	87.20	100.00	76.00	92.00	62.00	98.00	96.00	70.00	96.00	96.00	86.00
	Robot (Mosaic)	94.60	98.00	90.00	100.00	80.00	100.00	98.00	88.00	100.00	100.00	92.00
	Robot w/o Gripper (BG)	90.60	90.00	96.00	94.00	72.00	98.00	92.00	78.00	96.00	100.00	90.00
	Robot w/o Gripper (Black)	92.60	100.00	94.00	96.00	78.00	96.00	94.00	78.00	100.00	96.00	94.00
	Robot w/o Gripper (Mosaic)	96.40	92.00	100.00	100.00	92.00	98.00	98.00	90.00	100.00	100.00	94.00
	Background (Black)	72.20	86.00	98.00	34.00	52.00	96.00	96.00	56.00	22.00	100.00	82.00
	Background (Mosaic)	97.60	98.00	100.00	96.00	94.00	100.00	100.00	90.00	98.00	100.00	100.00
X-VLA	Baseline	97.20	100.00	100.00	94.00	96.00	98.00	92.00	100.00	100.00	98.00	94.00
	Target (BG)	83.80	100.00	94.00	76.00	98.00	96.00	4.00	84.00	100.00	98.00	88.00
	Target (Black)	84.40	100.00	100.00	94.00	90.00	92.00	0.00	86.00	100.00	88.00	94.00
	Target (Mosaic)	95.60	100.00	96.00	94.00	94.00	98.00	82.00	98.00	100.00	100.00	94.00
	Gripper (BG)	96.20	100.00	100.00	100.00	96.00	96.00	94.00	100.00	100.00	94.00	82.00
	Gripper (Black)	97.40	100.00	98.00	98.00	96.00	98.00	94.00	100.00	100.00	98.00	92.00
	Gripper (Mosaic)	98.40	100.00	100.00	98.00	96.00	98.00	98.00	100.00	100.00	100.00	94.00
	Robot (BG)	55.80	28.00	78.00	80.00	34.00	72.00	6.00	100.00	100.00	52.00	8.00
	Robot (Black)	93.40	96.00	98.00	92.00	98.00	100.00	62.00	100.00	100.00	96.00	92.00
	Robot (Mosaic)	95.20	98.00	98.00	96.00	96.00	98.00	74.00	100.00	100.00	98.00	94.00
	Robot w/o Gripper (BG)	92.40	96.00	100.00	92.00	94.00	100.00	52.00	100.00	100.00	98.00	92.00
	Robot w/o Gripper (Black)	94.00	94.00	98.00	98.00	98.00	100.00	58.00	100.00	100.00	100.00	94.00
	Robot w/o Gripper (Mosaic)	95.80	100.00	98.00	92.00	92.00	98.00	82.00	100.00	100.00	100.00	96.00
	Background (Black)	94.60	100.00	100.00	100.00	98.00	100.00	56.00	98.00	100.00	100.00	94.00
	Background (Mosaic)	98.00	100.00	100.00	100.00	94.00	98.00	92.00	100.00	100.00	100.00	96.00

Table 10: Success rates (%) across tasks and masking strategies on **LIBERO-Goal** for $\pi_{0.5}$, OpenVLA, OpenVLA-OFT, X-VLA. The instruction is provided in Tab. 15.

Setting	Avg	t1	t2	t3	t4
Baseline	79.17	79.17	70.83	95.83	70.83
Target(BG)	27.08	50.00	8.33	50.00	0.00
Target(Black)	44.79	50.00	33.33	95.83	0.00
Target(Mosaic)	72.92	66.67	70.83	87.50	66.67
Gripper(BG)	78.12	79.17	75.00	91.67	66.67
Gripper(Black)	75.00	83.33	66.67	91.67	58.33
Gripper(Mosaic)	75.00	83.33	70.83	91.67	54.17
Robot(BG)	76.04	75.00	70.83	95.83	62.50
Robot(Black)	76.04	87.50	70.83	87.50	58.33
Robot(Mosaic)	77.08	83.33	66.67	91.67	66.67
Background(BG)	59.38	83.33	0.00	100.00	54.17
Background(Black)	58.33	66.67	0.00	95.83	70.83
Background(Mosaic)	71.88	70.83	54.17	87.50	75.00

Table 11: Performance of X-VLA under different masking strategies on **Simpler**.

Model	Backbone	Size	Layers	Mask	Decoding	observation	Bench Coverage
$\pi_{0.5}$	PaliGemma	3B	18	Bi-dir	Flow Matching	RGB + Lang.	LIBERO, Calvin
X-VLA	Florence-2	0.9B	24	Bi-dir	Flow Matching	RGB + Lang. + Soft_P.	LIBERO, Clv, Simp., Rbt2
OpenVLA	Prismatic	7B	32	Causal	AutoRegressive	RGB + Lang.	LIBERO, Clv, Simpler
OpenVLA-OFT	Prismatic	7B	32	Bi-dir	L1 Regression	RGB + Lang.+ Prop.	LIBERO, Clv, RoboTwin2.0

Table 12: Overview of main and supplementary VLA model and simulator coverage. We conduct the full mechanistic diagnosis on $\pi_{0.5}$ and OpenVLA using LIBERO, and include X-VLA, OpenVLA-OFT, RoboTwin2.0, CALVIN, and Simpler as supplementary settings for broader validation. Lang. = language. LIB. = LIBERO; Cal. = CALVIN; Simp. = Simpler; Rbt2 = RoboTwin2.0.

Model	Stage	Region	Attention Mass	IoU top-10%	Peak Hit
OpenVLA	Phase 1	Gripper	0.2098 ± 0.0638	0.2198 ± 0.0702	0.4903
		Robot Arm	0.2406 ± 0.0592	0.1006 ± 0.0480	0.3131
		Object	0.2039 ± 0.0741	0.1374 ± 0.0902	0.2975
	Phase 2	Gripper	0.1930 ± 0.0616	0.2166 ± 0.0700	0.5148
		Robot Arm	0.2708 ± 0.0646	0.1209 ± 0.0468	0.3479
		Object	0.1965 ± 0.0748	0.1294 ± 0.0796	0.3389
	Full	Gripper	0.2014 ± 0.0632	0.2182 ± 0.0701	0.5026
		Robot Arm	0.2557 ± 0.0638	0.1108 ± 0.0485	0.3305
		Object	0.2475 ± 0.0863	0.1504 ± 0.0842	0.3599
$\pi_{0.5}$	Phase 1	Gripper	0.2076 ± 0.1064	0.2650 ± 0.1395	0.1887
		Robot Arm	0.1812 ± 0.0863	0.1008 ± 0.0768	0.0540
		Object	0.2213 ± 0.0868	0.1312 ± 0.0884	0.3798
	Phase 2	Gripper	0.1560 ± 0.0931	0.2057 ± 0.1324	0.1487
		Robot Arm	0.2042 ± 0.0649	0.1130 ± 0.0696	0.0372
		Object	0.2204 ± 0.0700	0.1421 ± 0.0793	0.4401
	Full	Gripper	0.1817 ± 0.1032	0.2353 ± 0.1392	0.1687
		Robot Arm	0.1927 ± 0.0772	0.1069 ± 0.0736	0.0456
		Object	0.2584 ± 0.0755	0.1379 ± 0.0738	0.4206

Table 13: LIBERO-10 attention grounding for gripper, robot arm, and object regions. All values are pooled over task-step samples. Gripper is the union of robot masks whose instance name contains "gripper"; Robot Arm is the union of the remaining robot masks; Object is the current subtask focus union. Full is recomputed over the full rollout using the union of both subtask focus sets.

$\pi_{0.5}$ evidence	Quantitative observation	Interpretation
Rollout contrast	Pretrained model: 0% success and 520-step failures; fine-tuned model: 100% success and 231-step completion on the representative rollout.	Finetuning changes attention from diffuse scene reading to executable manipulation tracking.
View allocation	In the layer-wise attention summary, action queries allocate roughly 19% mass to the wrist view and 9% to the agent view, after accounting for the BOS-dominated text sink.	$\pi_{0.5}$ uses the closer manipulation view more strongly, consistent with reliance on local contact and affordance cues.
Layer routing	Early layers L0–L3 mainly perform coarse input reading; visual use increases through L8–L13; L14 shows the strongest image–action coupling; L15–L17 mainly propagate action information.	The visual grounding signal is not monotonically stronger in later layers, but peaks near the layer where image removal is most damaging.
Attention sink	BOS self-attention reaches 0.81, and query groups send approximately 28–45% of their mass to BOS; other text tokens allocate about 70% mass within the text/BOS loop and only about 28% to images.	Raw text attention can overstate semantic language use; special tokens must be separated before interpreting instruction grounding.
Action slots	The ten action slots have nearly symmetric attention distributions under bidirectional self-attention.	$\pi_{0.5}$ encodes the action chunk as a coupled block, so visual routing affects the whole action segment rather than a single scalar action.

Table 14: Additional $\pi_{0.5}$ attention-routing evidence from the layer-wise visualization analysis. These statistics complement the IoU and mass results by explaining where the attention signal is routed inside the transformer.

Task Family	ID	Instruction	Capability
LIBERO-10 Long-horizon	t0	Put both the alphabet soup and tomato sauce into the basket.	Multi-object
	t1	Put both the cream cheese box and butter into the basket.	Multi-object
	t2	Turn on the stove and put the moka pot on it.	Sequential reasoning
	t3	Put the black bowl into the bottom drawer and close it.	Long-horizon
	t4	Put mugs onto corresponding plates.	Spatial planning
	t5	Place the book into the back compartment of the caddy.	Manipulation
	t6	Put mug on plate and pudding beside it.	Spatial reasoning
	t7	Put alphabet soup and cream cheese into basket.	Compositional
	t8	Put both moka pots on the stove.	Multi-object
	t9	Put mug into microwave and close it.	Long-horizon
LIBERO-Object Object-centric	t0	Pick up alphabet soup and place it in basket.	Object recognition
	t1	Pick up cream cheese and place it in basket.	Object grounding
	t2	Pick up salad dressing and place it in basket.	Visual grounding
	t3	Pick up BBQ sauce and place it in basket.	Recognition
	t4	Pick up ketchup and place it in basket.	Recognition
	t5	Pick up tomato sauce and place it in basket.	Recognition
	t6	Pick up butter and place it in basket.	Visual identification
	t7	Pick up milk and place it in basket.	Category generalization
	t8	Pick up chocolate pudding and place it in basket.	Recognition
	t9	Pick up orange juice and place it in basket.	Generalization
LIBERO-Spatial Spatial reasoning	t0	Pick bowl between plate and ramekin onto plate.	Relative position
	t1	Pick bowl next to ramekin onto plate.	Spatial reasoning
	t2	Pick bowl from table center onto plate.	Localization
	t3	Pick bowl on cookie box onto plate.	Spatial grounding
	t4	Pick bowl in top drawer onto plate.	3D localization
	t5	Pick bowl on ramekin onto plate.	Relative relation
	t6	Pick bowl beside cookie box onto plate.	Spatial grounding
	t7	Pick bowl on stove onto plate.	Spatial reasoning
	t8	Pick bowl beside plate onto plate.	Relative relation
	t9	Pick bowl on cabinet onto plate.	Localization
LIBERO-Goal Goal-directed	t0	Open the middle drawer of cabinet.	Goal execution
	t1	Put bowl on stove.	Goal reaching
	t2	Put wine bottle on cabinet.	Manipulation
	t3	Open top drawer and put bowl inside.	Multi-step planning
	t4	Put bowl on cabinet.	Goal completion
	t5	Push plate to front of stove.	Motion control
	t6	Put cream cheese into bowl.	Object interaction
	t7	Turn on stove.	Goal execution
	t8	Put bowl on plate.	Goal reaching
	t9	Put wine bottle on rack.	Target reaching
RoboTwin2.0 Dual-arm	t0	Click the alarm clock's center of the top side button on the table.	Button clicking
	t1	Click the bell's top center on the table.	Target clicking
	t2	Use both arms to grab the roller on the table.	Bimanual grasping
	t3	Use both arms to lift the pot.	Bimanual lifting
	t4	Use one arm to press the stapler.	Single-arm pressing
SimplerEnv Real-to-Sim	t0	Put carrot on plate.	Object placement
	t1	Put eggplant into yellow basket.	Goal manipulation
	t2	Put the spoon on the towel.	Spatial reasoning
	t3	Stack the green block on the yellow block.	Stacking manipulation

Table 15: Complete overview of LIBERO, RoboTwin2.0, Simpler benchmark tasks. LIBERO-10 evaluates long-horizon compositional manipulation, LIBERO-Object focuses on object-centric understanding, LIBERO-Spatial measures spatial reasoning, and LIBERO-Goal evaluates goal-conditioned control. RoboTwin2.0 subset (5/50 tasks) evaluates bimanual manipulation on five selected tasks. SimplerEnv-BridgeV2 tests real-to-sim transfer on Bridge-style manipulation tasks.

Task Family	ID	Instruction	Capability
Block Rotation	t0	Take the red block and rotate it to the right.	Block rotation
	t1	Take the red block and rotate it to the left.	Block rotation
	t2	Take the blue block and rotate it to the right.	Block rotation
	t3	Take the blue block and rotate it to the left.	Block rotation
	t4	Take the pink block and rotate it to the right.	Block rotation
	t5	Take the pink block and rotate it to the left.	Block rotation
Block Pushing	t6	Push the red block to the right.	Object pushing
	t7	Push the red block to the left.	Object pushing
	t8	Push the blue block to the right.	Object pushing
	t9	Push the blue block to the left.	Object pushing
	t10	Push the pink block to the right.	Object pushing
	t11	Push the pink block to the left.	Object pushing
Slider Drawer	t12	Push the sliding door to the left side.	Slider manipulation
	t13	Push the sliding door to the right side.	Slider manipulation
	t14	Pull the handle to open the drawer.	Drawer manipulation
	t15	Push the handle to close the drawer.	Drawer manipulation
Grasp & Lift	t16	Grasp and lift the red block.	Grasp and lift
	t17	Grasp and lift the blue block.	Grasp and lift
	t18	Grasp and lift the pink block.	Grasp and lift
Lift from Cabinet	t19	Lift the red block from the sliding cabinet.	Lift from container
	t20	Lift the blue block from the sliding cabinet.	Lift from container
	t21	Lift the pink block from the sliding cabinet.	Lift from container
Lift from Drawer	t22	Take the red block from the drawer.	Lift from drawer
	t23	Take the blue block from the drawer.	Lift from drawer
	t24	Take the pink block from the drawer.	Lift from drawer
Place Store	t25	Store the grasped block in the sliding cabinet.	Place into container
	t26	Store the grasped block in the drawer.	Place into drawer
Push into Drawer	t27	Slide the block until it falls into the drawer.	Push into drawer
Stack Unstack	t28	Stack the grasped block.	Stacking
	t29	Remove the stacked block.	Unstacking
Light Control	t30	Use the switch to turn on the light bulb.	Light control
	t31	Use the switch to turn off the light bulb.	Light control
	t32	Press the button to turn on the LED light.	LED control
	t33	Press the button to turn off the LED light.	LED control

Table 16: Overview of CALVIN manipulation tasks. CALVIN evaluates sequential multi-skill manipulation abilities, including object rotation, pushing, grasping, lifting, stacking, container interaction, drawer operation, and light control.