ENCODED BUT NOT IN CONTROL: REVEALING THE GROUNDING GAP IN VISION-LANGUAGE ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction following is central to language-conditioned robot policies: language should determine what to do when the same scene permits multiple valid actions. Yet successful execution alone cannot establish whether a policy follows the instruction or infers the task from the scene. We study this ambiguity through *scene-preserving instruction interventions*, using valid target substitutions, arbitrary nouns, and unrelated sentences while holding the scene fixed. We evaluate vision-language-action (VLA) policies and world-action models (WAMs) in simulation and in real-world experiments. Our analysis addresses three questions: ***(a)*** *Does task success imply instruction following?* When instructions request a different visible object, all evaluated policies predominantly approach and pick up the incorrect original target associated with the scene. ***(b)*** *Is this failure caused by language insensitivity?* Instruction perturbations affect task performance. A layerwise action lens shows intermediate action predictions respond to these perturbations. Linear probes accurately recover instructed targets, indicating modified instructions are encoded despite rarely determining target selection. ***(c)*** *Why does encoded language fail to control action?* Attention analysis indicates weak instruction-token contributions to action generation. Target-token attention can remain focused on the original object, revealing a mismatch between target encoding and visual grounding. UMAP and shared non-negative matrix factorization show target information remains accessible within representations increasingly organized by scene identity. Our findings expose a grounding gap concealed by nominal success and provide a diagnostic framework. They further establish a concrete criterion for progress: policies should reliably follow valid changes in user intent, even when they conflict with scene-favored behavior.

1 INTRODUCTION

Language-conditioned robot policies promise a natural interface for specifying what a robot should do in an environment. Recent VLAs (Intelligence et al., 2025; Bjorck et al., 2025) and WAMs (Yuan et al., 2026; Li et al., 2026b) demonstrate strong performance across manipulation tasks, embodiments, and visual environments, commonly evaluated by task success under a given instruction and environment. However, task success alone does not establish that the instruction determined the behavior. In many robot datasets, a scene is repeatedly paired with one target object or task, so the scene alone may suffice to infer the demonstrated behavior without resolving the requested behavior. A policy can consequently achieve high success by exploiting scene–task correlations while its ability to follow language remains unclear.

Instruction following requires evidence that changing a valid instruction in a fixed scene redirects behavior toward the newly specified target. Evaluations using linguistic perturbations and modality ablations reveal brittleness and task-dependent language sensitivity in VLA policies (Fei et al., 2025; Zhou et al., 2025b; Guo et al., 2026; Dong et al., 2026; Shi et al., 2026). More directly, LIBERO-CF and LangGap expose instruction-following failures under alternative feasible instructions in fixed scenes (Fang et al., 2026; Hou & Zhao, 2026), while InstructMove constructs scenes where language must distinguish among multiple feasible actions (Zhao et al., 2026). Mechanistic analyses further find that prompt information can remain decodable despite limited behavioral use, with ac-

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



Figure 1: **Target replacement reveals a gap in instruction following.** Holding the scene fixed, we change the instruction to request another visible and reachable object. In both LIBERO-Object and real-world manipulation, policies rarely grasp the currently instructed target (dashed green trajectories) and predominantly grasp the target named by the original instruction (solid red trajectories).

tion generation relying strongly on visual inputs (Grant et al., 2026). Building on these findings, we use *scene-preserving instruction interventions* to investigate how target-selection failures relate to the encoding, visual grounding, and contribution of instruction information to action generation, linking behavioral failures to action-level responses and internal representations across VLA and world-action models.

Part I: Does nominal task success imply instruction following? We revisit this question across VLA and WAM policies, with additional real-world experiments. When a valid instruction requests another object in the same scene, the policies predominantly approach and manipulate the original target. We refer to this directional preference as a *scene-conditioned behavioral prior*. It reveals a failure that standard success rates cannot expose: a policy may successfully reproduce the expected behavior for a familiar scene without allowing language to determine which behavior it executes.

Part II: Is target-conflict failure caused by language insensitivity? This failure cannot be attributed simply to absent linguistic influence. Controlled instruction perturbations produce systematic, task-dependent changes in policy performance. A layerwise action lens shows that intermediate action predictions respond differently to target substitutions and random sentences, evidencing language sensitivity in action generation. Linear probes also recover target and scene identities from fine-tuned representations with high accuracy, so target information remains available despite incorrect object selection. Policies thus respond to language at behavioral and representational levels, yet the failure persists even when the modified instruction affects action predictions and its target remains encoded—motivating a closer look at how this information contributes to action selection.

Part III: If target information is encoded, why does it fail to control action? Attention contribution analysis shows instruction tokens contribute little to action generation relative to visual inputs, and target-token attention can stay fixed on the original object even after the instruction names a different target. UMAP reveals a shift from object-based to scene-based clustering after fine-tuning; shared non-negative matrix factorization likewise identifies transitions from object-dominated to scene-dominated components in both $\pi_{0.5}$ and GR00T N1.7. High target decodability thus coexists with limited instruction contribution, misaligned visual attention, and increased scene dependence, characterizing why encoded language does not reliably guide target selection.

We describe this dissociation as a *representation-to-control grounding gap*: linguistic information can be represented within a policy without reliably governing action selection. Our analyses distinguish sensitivity to language, accessibility of target information, and behavioral adherence to the instruction. These target-selection failures also extend to the evaluated world-action models, showing that visual prediction objectives alone do not ensure instruction following. Our findings motivate a stricter grounding criterion: a valid change in instruction should reliably select a different behavior within the same scene. We hope this work encourages evaluations that test such control directly and provides a diagnostic basis for investigating where encoded instructions fail to guide action. Ultimately, progress should be measured by how reliably robots act on the user's current intent, even when it conflicts with familiar scene-task associations.

2 Related Work

Language-conditioned robot policies. Vision-language-action (VLA) models map visual observations and language instructions to robot actions. Recent models such as $\pi_{0.5}$ couple a pretrained vision-language backbone with a dedicated action expert that generates continuous action chunks

through flow matching (Intelligence et al., 2025). GR00T N1 adopts a dual-system architecture in which a vision-language module interprets visual observations and language instructions, followed by a diffusion transformer that generates continuous motor actions (Bjorck et al., 2025). World action models (WAMs) instead incorporate explicit visual-dynamics modeling into policy learning. LingBot-VA adopts an autoregressive diffusion framework that jointly learns future-frame prediction and policy execution, using a shared latent space and a Mixture-of-Transformers architecture to integrate visual and action tokens (Li et al., 2026b). Fast-WAM retains video co-training during training but removes explicit future prediction at test time, directly producing actions from the learned world representations (Yuan et al., 2026). These complementary architectures motivate our evaluation of $\pi_{0.5}$, GR00T N1.7, LingBot-VA, and FastWAM, allowing us to examine whether instruction-grounding failures persist across both VLA and world-action modeling paradigms.

Evaluating and improving instruction following. Recent studies investigate if task success reliably reflects language grounding. LIBERO-PRO reports persistent behaviors under corrupted instructions (Zhou et al., 2025b), while LIBERO-Plus evaluates robustness to linguistic, visual, and physical perturbations (Fei et al., 2025). Multilingual evaluation further reveals step-dependent language sensitivity (Dong et al., 2026). To test semantic control more directly, LIBERO-CF and LangGap vary instructions within existing scene layouts (Fang et al., 2026; Hou & Zhao, 2026), while InstructMove introduces semantic distractors that require language-dependent action selection (Zhao et al., 2026). Complementary methods strengthen language conditioning through counterfactual language–action augmentation in navigation (Glossop et al., 2026), Bayesian training objectives (Lian et al., 2026), counterfactual action guidance (Fang et al., 2026), and instruction augmentation coupled with guidance against a vision-only prior (Zhan et al., 2026). Building on these studies, we connect behavior under scene-preserving instruction interventions to internal policy representations. Target substitutions test semantic redirection, whereas arbitrary nouns and random sentences diagnose language dependence without necessarily specifying executable alternatives.

Language representations and behavioral influence. Probing studies distinguish decodable information from its use by a model, emphasizing the need for controls when interpreting probe accuracy (Belinkov, 2022; Hewitt & Liang, 2019). In robotics, Lu et al. (2025) decode symbolic object and action states from OpenVLA for monitoring within the cognitive architecture. Using counterfactual prompts, activation injection, sparse autoencoders, and probes, Grant et al. (2026) identify distinct VLM and action-expert roles and report decodable prompt categories despite weak behavioral sensitivity in $\pi_{0.5}$. VLA-Trace combines representation analysis, attention knockouts, and behavioral interventions to identify model-dependent control pathways and limitations in fine-grained semantic following (Shi et al., 2026). We build on these findings by pairing instruction interventions with layer-wise analyses of target and scene information across VLA and WAM policies, examining how instructed-target representations relate to action selection when language and scene-associated behavior conflict.

3 Experimental Settings

Environments and evaluation scope. Our experiments cover LIBERO (Spatial, Object, Goal, Long) (Liu et al., 2023), the clean and randomized settings of RoboTwin 2.0 (Chen et al., 2025), and six real-world manipulation scenes. These environments support a broad evaluation of instruction sensitivity. For controlled target-selection experiments, we use LIBERO-Object and real-world scenes containing multiple visible and reachable objects. These settings allow the instructed target to change while preserving the initial scene, manipulation primitive, and destination, enabling us to isolate target identity from other task requirements.

Instruction interventions. We consider three instruction conditions: Original, Noun Replacement, and Random Sentence. Original retains the nominal task instruction. Noun Replacement preserves the remaining instruction while modifying object expressions, and is instantiated in two ways: *In-scene Target Replacement*, which replaces only the manipulated target with another object present in the same scene, and *Random Noun*, which replaces all eligible object expressions in the instruction with randomly sampled nouns. Random Sentence replaces the complete instruction with a sentence generated independently. Details are provided in App. B.

Scene–target evaluation sets. For the representation analyses, we construct matched scene–target sets in LIBERO-Object and the real world. In LIBERO-Object, each of the ten evaluation scenes

Table 1: **Task completion under controlled target replacement.** The *current instructed target* is another object in the same scene requested by the modified instruction; the *original target* is named by the original instruction. "Tend" denotes a committed approach or manipulation attempt, and "Success" denotes successful task completion. All values are percentages (%). Policies predominantly retain the original target despite the modified instruction.

Environment	Model	Current instructed target		Original target	
		Tend	Success	Tend	Success
LIBERO-Object	$\pi_{0.5}$	13.0	11.5	86.5	69.5
	GR00T N1.7	1.3	0.5	96.5	90.9
	FastWAM	0.5	0.0	98.3	95.0
	LingBot-VA	1.8	1.3	96.3	84.4
Real World	$\pi_{0.5}$	10.8	5.8	80.8	74.2
	GR00T N1.7	4.2	3.3	94.2	90.8

contains six objects, the original target and five alternative objects present in the same scene, yielding $10 \times 6 = 60$ scene–target instances. Within each scene, the initial scene configuration and robot state are fixed, and only the instructed target changes. We construct an analogous real-world set with six scenes and six objects per scene, yielding $6 \times 6 = 36$ instances. The complete scene–target sets are provided in App. C.

Models. We evaluate two vision-language-action policies, $\pi_{0.5}$ (Intelligence et al., 2025) and GR00T N1.7 (Bjorck et al., 2025), and two world-action models, FastWAM (Yuan et al., 2026) and LingBot-VA (Li et al., 2026b). All four models are evaluated in simulation, while real-world experiments use $\pi_{0.5}$ and GR00T N1.7. Behavioral evaluations use closed-loop rollouts with frozen policy parameters. Model-specific observation interfaces, action spaces, episode horizons, action chunk length, instruction instances, and rollout counts are provided in the appendix.

4 Does Nominal Task Success Imply Instruction Following?

Guiding question Q1. *When the instruction requests another valid target in the same scene, does the policy follow the current instruction or retain the original target?*

Successful execution cannot distinguish instruction following from task inference when the scene and instruction suggest the same behavior. Following prior evaluations of alternative feasible instructions in fixed scenes (Fang et al., 2026; Hou & Zhao, 2026), we revisit this ambiguity through controlled target replacement. We change the instructed in the text prompt while preserving the initial scene and other robot configurations.

4.1 Target Replacement in LIBERO-Object and the Real World

We evaluate target replacement in LIBERO-Object using $\pi_{0.5}$, GR00T N1.7, FastWAM, and LingBot-VA, and in six real-world scenes using the two VLA policies. Alternative targets are visible and reachable; in LIBERO-Object, each manipulable object is also represented in the training set. We report committed approaches or manipulation attempts (*Tend*) and successful task execution (*Success*). *Tend* records whether the policy makes a committed approach or manipulation attempt toward a target, whereas *Success* requires successfully finish the task.

Across the four models in LIBERO-Object, 86.5%–98.3% of trials are directed toward the original target, compared with 0.5%–13.0% toward the current instructed target (Tab. 1). Original-target task completion reaches 69.5%–95.3%, while current-target completion remains at 0.0%–11.5%. The same preference persists in physical execution: original-target pickup rates are 74.2% and 90.8% for $\pi_{0.5}$ and GR00T N1.7, respectively, compared with 5.8% and 3.3% for the current target.

These results corroborate previously reported counterfactual failures (Fang et al., 2026; Hou & Zhao, 2026) and extend the evidence to the evaluated world-action models. The policies frequently ma-

Table 2: **Task performance under instruction corruption.** Main values denote success rates (%); colored subscripts indicate relative decreases (↓) or increases (↑) from the original-instruction baseline (%). The results demonstrate that instruction corruption affects performance, with sensitivity varying across tasks and evaluation conditions.

Benchmark	Task / Setting	Instruction	VLA		WAM	
			$\pi_{0.5}$	GR00T N1.7	FastWAM	LingBot-VA
LIBERO	Spatial	Original	98.8	97.7	98.2	98.5
		Random noun	99.0$_{\uparrow 0.2}$	96.6$_{\downarrow 1.1}$	73.6$_{\downarrow 25.1}$	90.4$_{\downarrow 8.2}$
		Random sentence	66.4$_{\downarrow 32.8}$	53.2$_{\downarrow 45.5}$	37.4$_{\downarrow 61.9}$	75.2$_{\downarrow 23.7}$
	Object	Original	98.2	98.5	100.0	99.6
		Random noun	96.6$_{\downarrow 1.6}$	94.4$_{\downarrow 4.1}$	92.6$_{\downarrow 7.4}$	83.4$_{\downarrow 16.3}$
		Random sentence	64.9$_{\downarrow 33.9}$	52.8$_{\downarrow 46.4}$	87.0$_{\downarrow 13.0}$	49.2$_{\downarrow 50.6}$
	Goal	Original	98.0	97.5	97.0	97.2
		Random noun	81.5$_{\downarrow 16.8}$	75.2$_{\downarrow 22.9}$	49.0$_{\downarrow 49.5}$	61.4$_{\downarrow 36.8}$
		Random sentence	11.7$_{\downarrow 88.1}$	4.0$_{\downarrow 95.9}$	4.8$_{\downarrow 95.1}$	12.0$_{\downarrow 87.7}$
	Long	Original	92.4	94.4	95.2	98.5
		Random noun	94.2$_{\uparrow 2.0}$	87.4$_{\downarrow 7.4}$	82.2$_{\downarrow 13.7}$	89.6$_{\downarrow 9.0}$
		Random sentence	78.6$_{\downarrow 14.9}$	59.6$_{\downarrow 36.8}$	47.6$_{\downarrow 50.0}$	79.8$_{\downarrow 19.0}$
RoboTwin 2.0	Clean	Original	72.8	69.5	91.9	92.9
		Random noun	60.3$_{\downarrow 17.2}$	58.2$_{\downarrow 16.2}$	81.6$_{\downarrow 11.2}$	82.6$_{\downarrow 11.1}$
		Random sentence	48.9$_{\downarrow 32.9}$	48.4$_{\downarrow 30.4}$	51.0$_{\downarrow 44.5}$	59.6$_{\downarrow 35.8}$
	Randomized	Original	71.3	67.4	91.8	91.6
		Random noun	56.1$_{\downarrow 21.3}$	48.9$_{\downarrow 27.4}$	76.6$_{\downarrow 16.5}$	76.4$_{\downarrow 16.6}$
		Random sentence	47.0$_{\downarrow 34.2}$	42.1$_{\downarrow 37.5}$	38.8$_{\downarrow 57.7}$	48.6$_{\downarrow 46.9}$

nipulate the object that is no longer requested, as illustrated in Fig. 1, motivating closer analysis of how the modified instruction is processed.

Answer to Q1. ***Nominal task success does not establish instruction following.*** *Across the evaluated policies, valid target replacement predominantly leaves behavior directed toward the original target, despite successful manipulation.*

5 Is This Failure Caused by Language Insensitivity?

Guiding question Q2. *Do policies remain sensitive to language and encode the currently instructed target despite failing to follow it?*

The original-target preference in Sec. 4 does not establish whether the policies ignore the modified instruction. We examine language sensitivity at three levels: changes in task success under instruction perturbations, changes in intermediate action predictions measured by a layerwise action lens, and target decodability measured by linear probes. These analyses test whether instruction-following failures persist even when language affects action prediction and the currently instructed target remains recoverable from the policy's representations.

5.1 Instruction Perturbations Reveal Behavioral Sensitivity

We evaluate the instruction-corruption conditions defined in Sec. 3: Original, Any Noun, and Random Sentence. All four policies are evaluated on LIBERO and RoboTwin 2.0 under clean and randomized settings, with success measured against the original task.

LIBERO: Random sentences reduce success across all four models and evaluated suites (Tab. 2). Noun replacement generally produces smaller, task-dependent reductions. For $\pi_{0.5}$ and GR00T N1.7, its effect is limited on Spatial, Object, and Long, but larger on Goal. Both world-action

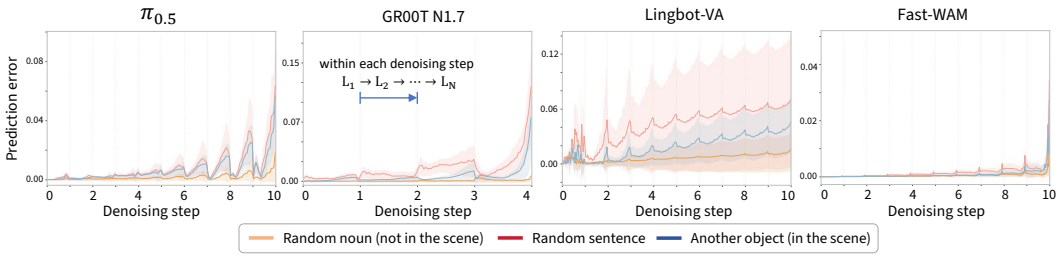


Figure 2: **Layerwise action predictions under instruction interventions.** Curves show mean squared deviations from original-instruction predictions at matched layers and denoising steps. Random sentences (red) generally produce the largest deviations, followed by in-scene target replacements (blue) and out-of-scene nouns (orange), indicating language sensitivity within generation.

models likewise exhibit their largest noun-replacement reductions on LIBERO-Goal. **RoboTwin:** Both interventions reduce success for every model in the clean and randomized settings. Random sentences consistently have a larger effect than noun replacement. These results show that language affects task performance, with sensitivity varying across tasks and evaluation conditions.

5.2 Instruction Changes Affect Intermediate Action Predictions

Inspired by the logit lens (nostalgebraist, 2020), we examine action-generation sensitivity using a layerwise action lens. At each layer l and denoising step s, we apply the policy's action readout to intermediate hidden states in normalized action coordinates. These readouts diagnose intermediate representations rather than provide calibrated predictions of executed actions. For the same visual observation, we compare each intervention q with the original instruction at the same layer and step:

$$D_q^{(s,l)} = \frac{1}{d} \left\| \hat{\mathbf{a}}_q^{(s,l)} - \hat{\mathbf{a}}_{\text{orig}}^{(s,l)} \right\|_2^2, \tag{1}$$

where d is the number of actions within an episode. We compare noun replacement with an out-of-scene noun or another visible object, and complete instruction replacement with a random sentence. Details are provided in App. E.

Across all four models, Fig. 2 shows a consistent overall ordering: out-of-scene nouns yield the smallest deviations, followed by in-scene replacements and random sentences. Separation generally increases toward later denoising steps, with variation across layers and models. These results show that instruction changes affect intermediate action predictions, complementing their effects on task performance in Sec. 5.1. However, deviation does not measure movement toward the instructed target: rollouts in Sec. 4 predominantly retain the original target. We next test whether the current target's identity remains recoverable from internal representations.

5.3 The Current Target Remains Linearly Decodable

We probe whether the currently instructed target remains accessible in the internal representations of $\pi_{0.5}$ and GR00T N1.7 before and after fine-tuning. Using the matched scene–target sets from LIBERO-Object and the real world, we train separate linear probes to predict target identity and scene identity from instruction representations. Full feature-extraction, label-construction, training, and evaluation details are provided in App. D. Fig. 3 shows near-ceiling decodability in both models and environments. On a test set, target identity is recovered with essentially perfect accuracy before and after fine-tuning, while scene identity is also decoded with near-ceiling accuracy.

The currently instructed target therefore remains linearly accessible despite the unreliable target selection observed in Sec. 4. This rules out complete loss of target identity from the probed representations as an explanation for the behavioral failure.

Answer to Q2. ***The failure cannot be explained simply by language insensitivity or the absence of target information.*** *Instruction perturbations affect performance, and the current target remains highly decodable in the probed policies, despite their unreliable target following.*

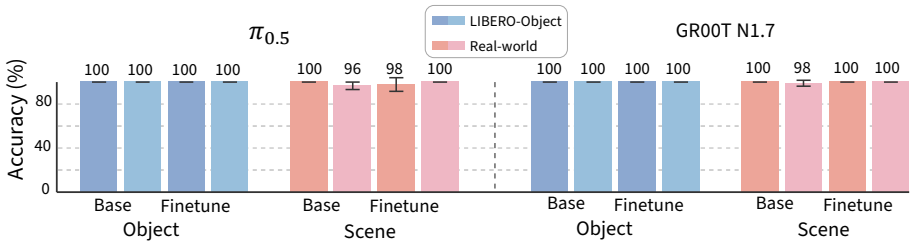


Figure 3: **Linear probing of target and scene identity.** Probes on mean-pooled instruction-token representations achieve near-ceiling accuracy in $\pi_{0.5}$ and GR00T before and after fine-tuning on LIBERO-Object and real-world data, showing that both identities remain accessible despite instruction-following failures.

6 Why Does Encoded Language Fail to Control Action?

Guiding question Q3. *If the current target remains encoded, how can visual attention and action generation remain misaligned with the instruction, and what changes in representational organization accompany this failure?*

Sec. 5 shows that instruction changes affect task performance and intermediate action predictions, while the current target remains linearly decodable in the probed policies. We now examine how this information participates in action generation and relates to visual grounding. We first quantify the relative contribution of instruction tokens to action generation. Then UMAP and shared NMF characterize how target- and scene-dependent organization changes after fine-tuning. These analyses provide complementary diagnostics of the gap between target encoding and behavioral control.

6.1 Instruction Tokens Contribute Weakly to Action Generation

Target decodability does not establish how strongly instruction tokens contribute to action generation. We quantify their relative contribution to the action expert by aggregating attention-weighted value vectors after output projection (Kobayashi et al., 2020; Song et al., 2026). For action-query token i and source token j, we compute

$$e_{ij} = \sum_{\ell=1}^{L} \sum_{h=1}^{H_\ell} A^{\ell}_{hij} \left(v^{\ell}_{hj} W^{O,\ell}_{h} \right), \qquad s_{ij} = \frac{\|e_{ij}\|_2}{\sum_{j'} \|e_{ij'}\|_2}. \tag{2}$$

where A^{ℓ}_{hij} is the attention probability, v^{ℓ}_{hj} the value vector, and $W^{O,\ell}_{h}$ the head's output projection. Normalization follows aggregation across all layers and heads. We then sum s_{ij} by source modality and aggregate over active action-query tokens.

Instruction tokens account for a smaller share of the measured contribution than the dominant visual or state inputs (Fig. 4). For $\pi_{0.5}$, this share is 25.6%–27.9% on LIBERO, approximately 1.7% on RoboTwin 2.0, and 9.7% in the real world. The corresponding values for GR00T N1.7 are 18.5%–23.4%, approximately 12%, and 13.5%. Thus, high target decodability coexists with a relatively small instruction contribution under this attention-based measure.

6.2 Target Replacement Does Not Reliably Redirect Visual Attention

We examine whether attention from instructed-target tokens aligns with the corresponding object using the LIBERO-Object evaluation set Fig. 5 shows a representative $\pi_{0.5}$ example in which the instructed target changes from *tomato sauce* to *BBQ sauce*.

This observation reveals a mismatch between target encoding and visual grounding: the current target remains decodable, yet target-token attention can favor the original object. This pattern accompanies the behavioral failures in Sec. 4, without establishing a causal role for attention. We next examine whether fine-tuning also changes how instruction representations are organized around target and scene identity.

6.3 Fine-Tuning Shifts Instruction Representations toward Scene Identity

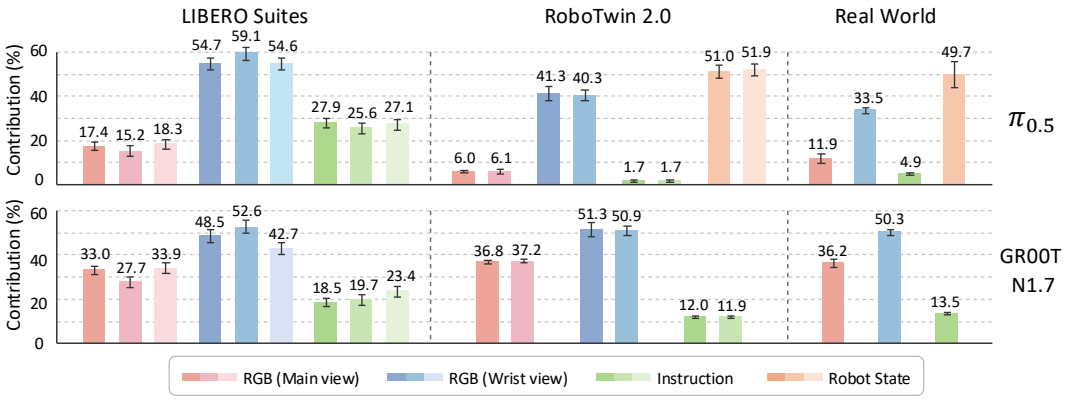


Figure 4: **Modality contributions to action generation.** Instruction tokens contribute relatively weakly to action generation in both $\pi_{0.5}$ and GR00T N1.7, with visual or robot-state inputs accounting for larger shares across simulation and real-world settings.

We examine how fine-tuning changes the organization of instruction representations in $\pi_{0.5}$. Using the matched LIBERO-Object scene–target evaluation set, we visualize layer-wise instruction representations from the base and fine-tuned checkpoints with UMAP. Details are provided in App. F.

As shown in Fig. 6, before fine-tuning, the projections show clear target-based clustering across the inspected layers. Fine-tuning substantially reorganizes this geometry. The early layer retains clear target-based clustering, intermediate layers become predominantly organized by scene identity, and the final layers are more dispersed while retaining a tendency toward target-based grouping. Together with the probing results, these visualizations suggest that fine-tuning changes how target information is organized, without erasing it.

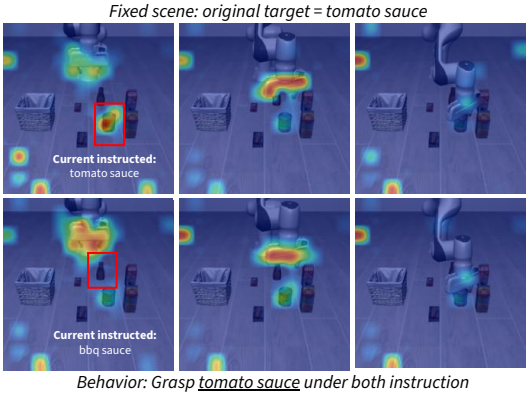


Figure 5: **Target-token visual attention under target replacement.** When the instructed target changes from tomato sauce to BBQ sauce, attention remains focused on the original tomato sauce, which the policy grasps under both instructions. Red boxes mark the currently instructed target.

6.4 Shared Semi-NMF Reveals a Shift toward Scene-Dependent Organization

We quantify the reorganization observed in UMAP using shared semi-NMF (Ding et al., 2008). We jointly factorize matched base and fine-tuned instruction representations:

$$\begin{bmatrix} X_{\text{base}} \\ X_{\text{FT}} \end{bmatrix} \approx \begin{bmatrix} A_{\text{base}} \\ A_{\text{FT}} \end{bmatrix} B^{\top}, \qquad A_{\text{base}}, A_{\text{FT}} \geq 0, \tag{3}$$

where the shared signed basis B aligns components across checkpoints, and A contains their condition-level activations.

For each component, we fit scene-only, target-only, and additive scene–target models. Their explained variances, R^2_S, R^2_T, and R^2_{ST}, yield the unique contributions of scene and target identity:

$$U_{\text{scene}} = R^2_{ST} - R^2_T, \qquad U_{\text{target}} = R^2_{ST} - R^2_S. \tag{4}$$

Each term measures the additional variance explained by one factor after controlling for the other. Implementation and variance-partitioning details are provided in App. G.

In the $(U_{\text{target}}, U_{\text{scene}})$ plane, all 12 components shift upper-left at every analyzed layer from layer 2 onward, indicating increased scene-unique and decreased target-unique variance after fine-tuning.

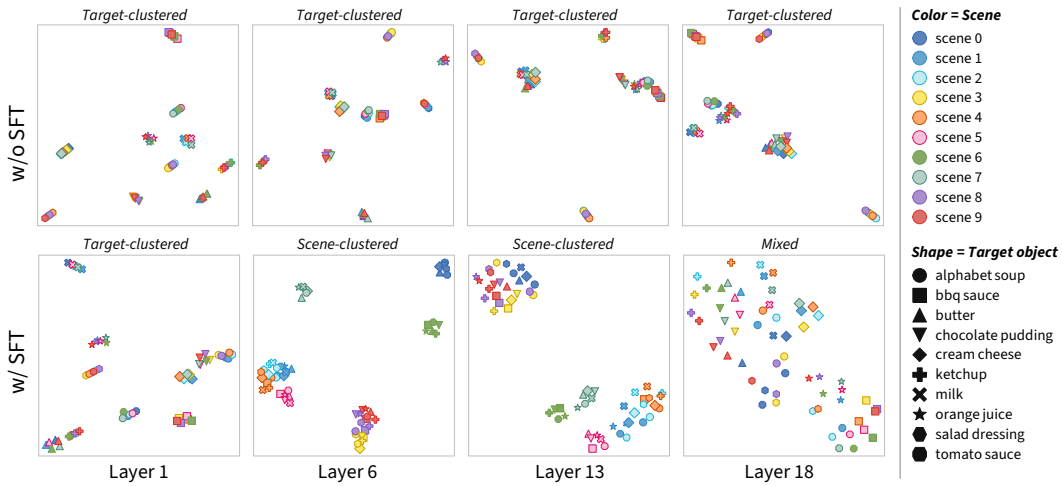


Figure 6: **UMAP projections of $\pi_{0.5}$ instruction representations.** Columns show layers 1, 6, 13, and 18; rows compare Base and SFT. Shape denotes target identity; color denotes scene. Base representations cluster by target. After fine-tuning, the first layer retains target clustering, intermediate layers group by scene, and the final layer is more dispersed with residual target grouping.

Representative results in Fig. 7 illustrate the shift from target-dominated to scene-dominated organization in LIBERO and the real-world setting. Preserved target information therefore coexists with scene-dominated organization.

Although both scene and target identities remain linearly decodable (as described in Sec. 5.3), fine-tuning shifts the shared components from target-dominated toward scene-dominated organization. Target information is preserved, but becomes less prominent in the variation captured by these components. This helps reconcile high probe accuracy with instruction-following failures and is consistent with the preference for the original target, without establishing a causal link.

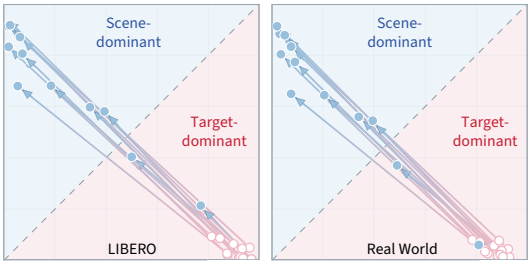


Figure 7: **Shared semi-NMF components before and after fine-tuning.** Arrows connect components before (open circles) and after fine-tuning (filled circles). Across both settings, components shift from target-dominated to scene-dominated organization, with greater scene-unique and lower target-unique variance.

Answer to Q3. ***Target information remains encoded, but its participation in action generation and alignment with the instructed object are limited.*** *Instruction tokens account for a relatively small share of the measured attention contribution to action generation, target-token visual attention can remain focused on the original object, and fine-tuning shifts representational organization toward scene identity. These findings characterize how preserved target information coexists with unreliable instruction following, without establishing a causal mechanism for the failure.*

7 Conclusion

We investigate instruction grounding through a progression from behavioral failures to language sensitivity, target encoding, and participation in action generation. Across the evaluated policies, changed instructions affect computation and preserve recoverable target information, yet often fail to redirect target selection. The internal analyses reveal weak instruction contributions, persistent attention to the original object, and a shift toward scene-dependent representational organization. This evidence places the use of encoded language at the center of the grounding problem. It motivates evaluation and training that explicitly require instructions to select among competing valid behaviors within the same scene.

AI USE STATEMENT

We used ChatGPT to polish the manuscript for clarity, grammar, and conciseness. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work examines instruction-following failures in robot policies, with implications for reliable human–robot interaction. Our findings highlight the need to verify behavioral adherence to user instructions before deployment.

REPRODUCIBILITY STATEMENT

The main text and appendix describe the evaluated models, instruction interventions, experimental settings, and analysis procedures. We provide implementation details and evaluation protocols to support reproduction of the reported results.

REFERENCES

Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 22304–22325, 2023.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113, 2026.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9490–9498. IEEE, 2025.

Jiahang Cao, Qiang Zhang, Hanzhong Guo, Jiaxu Wang, Hao Cheng, and Renjing Xu. Modality-composable diffusion policy via inference-time distribution-level composition. *arXiv preprint arXiv:2503.12466*, 2025a.

Jiahang Cao, Qiang Zhang, Jingkai Sun, Jiaxu Wang, Hao Cheng, Yulin Li, Jun Ma, Kun Wu, Zhiyuan Xu, Yecheng Shao, et al. Mamba policy: Towards efficient 3d diffusion policy with hybrid selective state models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11359–11366. IEEE, 2025b.

Jiahang Cao, Yize Huang, Hanzhong Guo, Qiang Zhang, Rui Zhang, Weijian Mai, Mu Nan, Jiaxu Wang, Hao Cheng, Jingkai SUN, et al. Compose your policies! improving diffusion-based or flow-based robot policies via test-time distribution-level composition. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 152825–152879, 2026.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, et al. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025a.

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

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Kaiyuan Chen, Shuangyu Xie, Letian Fu, Justin Yu, William Pacini, Sandeep Bajamahal, Hudson Kim, Jaimyn Drake, Daehwa Kim, Haoru Xue, et al. Gap: A graph-as-policy multi-agent self-learning harness for variational automation tasks. *arXiv preprint arXiv:2607.05369*, 2026.

Lili Chen, Shikhar Bahl, and Deepak Pathak. Playfusion: Skill acquisition via diffusion from language-annotated play. In *Conference on Robot Learning (CoRL)*, pp. 2012–2029. PMLR, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Dongzhou Cheng, Taoran Yi, Ye Fang, Xingwu Zhang, Fan Feng, Yixuan Li, Gengxiong Zhuang, Rongze Wang, Shuai Yang, Wei Song, et al. In-context robot learning with vlm agents. *arXiv preprint arXiv:2609.19138*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research (IJRR)*, 44(10-11):1684–1704, 2025.

Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 32(1):45–55, 2008.

Xuan Dong, Zhe Han, Tianhao Niu, Qingfu Zhu, and Wanxiang Che. When does language matter? multilingual instructions reveal step-wise language sensitivity in vision-language-action models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 44615–44629, 2026.

Maximilian Du and Shuran Song. Dynaguide: Steering diffusion polices with active dynamic guidance. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:44192–44221, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:9156–9172, 2023.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in vlas. *arXiv preprint arXiv:2602.17659*, 2026.

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

Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025a.

Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *arXiv preprint arXiv:2502.16707*, 2025b.

Liane Galanti, Dhruv Shah, and Tri Dao. Addressing the orchestration gap in generalist robots via physical agency. *arXiv preprint arXiv:2607.21725*, 2026.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 100903–100933, 2026a.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026b.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. Cast: Counterfactual labels improve instruction following in vision-language-action models. *IEEE Robotics and Automation Letters (RAL)*, 2026.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models. *arXiv preprint arXiv:2603.19233*, 2026.

Jianing Guo, Zhenhong Wu, Chang Tu, Yiyao Ma, Xiangqi Kong, Zhiqian Liu, Jiaming Ji, Shuning Zhang, Yuanpei Chen, Kai Chen, et al. On robustness of vision-language-action model against multi-modal perturbations. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 70248–70272, 2026.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019.

Sigmund H Høeg, Yilun Du, and Olav Egeland. Streaming diffusion policy: Fast policy synthesis with variable noise diffusion models. *arXiv preprint arXiv:2406.04806*, 2024.

Yuchen Hou and Lin Zhao. Langgap: Diagnosing and closing the language gap in vision-language-action models. *arXiv preprint arXiv:2603.00592*, 2026.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Mengzhao Jia, Yang Lin, Xixin Zhang, Zhihan Zhang, Xiaobai Liu, and Meng Jiang. Agent as policy for robotic manipulation. *arXiv preprint arXiv:2609.12541*, 2026.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, 2020.

Chenxu Li, Zixuan Chen, Yetao Li, Jiapeng Xu, Hongyu Ding, Jieqi Shi, Jing Huo, and Yang Gao. Momastage: Skill-state graph guided planning and closed-loop execution for long-horizon indoor mobile manipulation. *arXiv preprint arXiv:2603.08383*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Ruiying Li, Yunlang Zhou, YuYao Zhu, Kylin Chen, Jingyuan Wang, Sukai Wang, Kongtao Hu, Minhui Yu, Bowen Jiang, Zhan Su, et al. Roboclaw: An agentic framework for scalable long-horizon robotic tasks. *arXiv preprint arXiv:2603.11558*, 2026c.

Xiang Li, Varun Belagali, Jinghuan Shang, and Michael S Ryoo. Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16841–16849. IEEE, 2024.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. Langforce: Bayesian decomposition of vision language action models via latent action queries. *arXiv preprint arXiv:2601.15197*, 2026.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 88446–88463, 2026.

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

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 29982–30009, 2025b.

Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, Xiao Ouyang, Hang Su, and Jun Zhu. Rdt2: Exploring the scaling limit of umi data towards zero-shot cross-embodiment generalization. *arXiv preprint arXiv:2602.03310*, 2026.

Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2(4), 2022.

Yijun Liu, Yuwei Liu, Yuan Meng, Jieheng Zhang, Yuwei Zhou, Ye Li, Jiacheng Jiang, Kangye Ji, Shijia Ge, Zhi Wang, et al. Spatial policy: Guiding visuomotor robotic manipulation with spatial-aware modeling and reasoning. *arXiv preprint arXiv:2508.15874*, 2025c.

Yuejiang Liu, Jubayer Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 4594–4627, 2025d.

Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture. In *IEEE International Conference on AI and Data Analytics (ICAD)*, pp. 1–8. IEEE, 2025.

Yiyang Lu, Yufeng Tian, Zhecheng Yuan, Xianbang Wang, Pu Hua, Zhengrong Xue, and Huazhe Xu. H^3dp: Triply-hierarchical diffusion policy for visuomotor learning. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 13237–13261, 2026.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18081–18090. IEEE, 2024.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 36352–36372, 2026.

nostalgebraist. Interpreting gpt: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *arXiv preprint arXiv:2407.05996*, 2024.

Vaibhav Saxena, Yotto Koga, and Danfei Xu. Constrained-context conditional diffusion models for imitation learning. *arXiv preprint arXiv:2311.01419*, 2023.

SP Sharan, Ruihan Zhao, Zhangyang Wang, Sandeep P Chinchali, et al. Plan diffuser: Grounding llm planners with diffusion models for robotic manipulation. In *Bridging the Gap between Cognitive Science and Robot Learning in the Real World: Progresses and New Directions*, 2024.

Haoyuan Shi, Xiancong Ren, Yingji Zhang, Qinfan Zhang, Jiayu Hu, Haozhe Shan, Han Dong, Jinpeng Lu, Yinda Chen, Yi Zhang, et al. Vla-trace: Diagnosing vision-language-action models through representation and behavior tracing. *arXiv preprint arXiv:2605.30117*, 2026.

Alan Z Song, Yinjie Chen, Mu Nan, Rui Zhang, Jiahang Cao, Weijian Mai, Muquan Yu, Hossein Adeli, Deva Ramanan, Michael J Tarr, et al. Elastic attention cores for scalable vision transformers. *arXiv preprint arXiv:2605.12491*, 2026.

Jiayi Su, Yixin Zheng, Mi Yan, Li Yi, Zhizheng Zhang, and He Wang. GPT 6 Astra as an embodied policy. Technical report and code, 2026. URL https://github.com/anonymous-report-421/eval-of-gpt-6-astra-as-policy.

Zhanyi Sun and Shuran Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:174280–174305, 2026.

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

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

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

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

Vitalis Vosylius, Younggyo Seo, Jafar Uruç, and Stephen James. Render and diffuse: Aligning image and action spaces for diffusion-based behaviour cloning. *arXiv preprint arXiv:2405.18196*, 2024.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: toward fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters (RAL)*, 10(4):3988–3995, 2025a.

Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In *International Conference on Machine Learning (ICML)*, 2025b.

Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *IEEE Robotics and Automation Letters (RAL)*, 9(12):11834–11840, 2024.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 10641–10662, 2024.

Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Annual Conference on Robot Learning (CoRL)*, 2023.

Peiran Xu, Jiaqi Zheng, and Yadong Mu. Roboagent: Chaining basic capabilities for embodied task planning. *arXiv preprint arXiv:2604.07774*, 2026.

Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, et al. Maniflow: A general robot manipulation policy via consistency flow training. *arXiv preprint arXiv:2509.01819*, 2025.

Quantao Yang, Michael C Welle, Danica Kragic, and Olov Andersson. S^2-diffusion: Generalizing from instance-level to category-level skills in robot manipulation. *IEEE Robotics and Automation Letters (RAL)*, 2025.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 28213–28239, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3d diffusion policies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2873–2880. IEEE, 2025.

Zhihao Zhan, Yuhao Chen, Jiaying Zhou, Qinhan Lyu, Hao Liu, Keze Wang, Liang Lin, and Guangrun Wang. Stable language guidance for vision–language–action models. In *Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, pp. 4137–4159, 2026.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:24195–24228, 2026.

Xinyu Zhang, Yuhan Liu, Haonan Chang, Liam Schramm, and Abdeslam Boularias. Autoregressive action sequence learning for robotic manipulation. *IEEE Robotics and Automation Letters (RAL)*, 10(5):4898–4905, 2025.

Mengao Zhao, Ziang Li, Chaodong Huang, Mengchen Ma, Haoyi Jiang, Yiwei Jin, Xinjie Wang, Yun Du, Xuewu Lin, Taojun Ding, et al. Instructmove: A text-indispensable benchmark for instruction-following manipulation. *arXiv preprint arXiv:2608.22990*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2goal: From world model to general goal-conditioned policy. *arXiv preprint arXiv:2512.23541*, 2025a.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

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

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, et al. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. In *Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395, 2025c.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

APPENDIX

1. General Experimental Settings – Model sources and implementations, benchmark descriptions, and evaluation protocols.
2. Instruction Intervention Details – Construction of NOUN REPLACEMENT (*In-scene Target Replacement* and *Random Noun*) and RANDOM SENTENCE.
3. Details on Scene–Target Evaluation Sets – Construction of the matched LIBERO-Object and real-world scene–target sets, including scenes, instructed-target sets, and instruction templates.
4. Linear Probe Settings – Scene- and target-classification tasks, feature extraction, one-hot labels, and ridge-regression protocol.
5. Layer-wise Action Lens Analysis – Paired counterfactual replay, model-specific intermediate action readouts, and MSE-based comparison.
6. NMF Analysis – Factorization setup, representation preprocessing, shared components, and quantitative analysis.
7. Literature Review – Detailed literature review on Vision–action models, vision–language–action models, world–action models, and agentic robotics.

A GENERAL EXPERIMENTAL SETTINGS

A.1 MODEL SOURCES AND IMPLEMENTATIONS

We evaluate four language-conditioned robot policies two model families: two vision-language-action (VLA) policies, $\pi_{0.5}$ (Intelligence et al., 2025) and GR00T N1.7 (Bjorck et al., 2025), and two world-action models (WAMs), FastWAM (Yuan et al., 2026) and LingBot-VA (Li et al., 2026b). All behavioral evaluations are conducted in closed loop with frozen policy parameters. We retain the observation preprocessing, action representation, and inference interface of each public implementation rather than converting all models into a shared policy architecture.

Source repositories. The model implementations are based on the following public repositories:

- $\pi_{0.5}$: Physical Intelligence OpenPI (GitHub).
- GR00T N1.7: NVIDIA Isaac-GR00T (GitHub).
- FastWAM (GitHub).
- LingBot-VA (GitHub).
- XPolicyLab (GitHub), used for our RoboTwin training pipeline for $\pi_{0.5}$ and GR00T N1.7.

A.2 LIBERO BENCHMARK

LIBERO (Liu et al., 2023) is a language-conditioned manipulation benchmark designed to study knowledge transfer across manipulation tasks. We evaluate the four commonly used ten-task suites: LIBERO-SPATIAL, LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-LONG.

The suites emphasize different forms of generalization. SPATIAL varies spatial relations while largely preserving object semantics; OBJECT varies the manipulated objects; GOAL varies task

goals; and LONG contains longer-horizon, multi-stage manipulation tasks. Each task is associated with a natural-language instruction and a benchmark-defined success predicate.

The LIBERO simulator uses a Franka Panda robot and provides RGB observations from an external agent-view camera and an eye-in-hand wrist camera together with robot proprioception. The low-level control action is seven-dimensional, consisting of end-effector translation, end-effector rotation, and gripper control. We use the benchmark-provided initial states and native task-success predicates.

For all instruction-intervention experiments, the initial environment state and robot state are held fixed across instruction conditions. Thus, the ORIGINAL, NOUN REPLACEMENT, and RANDOM SENTENCE conditions differ only in the language input.

A.3 LIBERO EVALUATION PROTOCOLS

Because the four policies expose different action-chunking and inference interfaces, we retain the evaluation protocol associated with each public implementation. For each model, we use a checkpoint fine-tuned on the full LIBERO benchmark. When such a checkpoint is not publicly available, we fine-tune the model ourselves following the official training guidelines and recommended settings. Importantly, the evaluation protocol for a given model is kept identical across all instruction conditions.

A.4 ROBOTWIN 2.0 BENCHMARK

RoboTwin 2.0 (Chen et al., 2025) is a simulation benchmark for generalizable bimanual robotic manipulation. We evaluate all four policies on the 50-task benchmark using the ALOHA-AgileX dual-arm embodiment.

We consider two evaluation conditions. The CLEAN condition uses the nominal task configuration, whereas the RANDOMIZED condition introduces benchmark-defined domain randomization. RoboTwin 2.0 randomizes multiple visual and physical factors, including background and tabletop textures, scene clutter, illumination, tabletop height, and camera viewpoint. These variations test whether a policy trained on demonstrated manipulation behaviors remains robust to changes in the surrounding scene.

The RoboTwin environment provides observations from a global head camera and two wrist cameras together with the bimanual robot state. Policies predict actions for the ALOHA-AgileX embodiment through their corresponding RoboTwin policy adapters.

A.5 ROBOTWIN TRAINING AND EVALUATION PROTOCOL

For $\pi_{0.5}$ and GR00T N1.7, the RoboTwin checkpoints used in this paper are trained by us following the default training pipelines provided by XPolicyLab. FastWAM and LingBot-VA are evaluated through their corresponding public RoboTwin implementations.

RoboTwin 2.0 uses task-specific evaluation horizons rather than a single global maximum episode length. In evaluation mode, each task reads its control-step limit from the benchmark configuration. We retain these benchmark-defined limits.

Following the benchmark evaluation protocol, each task is evaluated with 100 rollouts. An episode terminates when either the task-specific success condition is satisfied or its corresponding control-step limit is reached. The benchmark success predicate is used directly without introducing an additional task-completion heuristic.

Within each model, the same simulator seeds, task configurations, episode limits, observation preprocessing, and action-execution parameters are used for the ORIGINAL, NOUN REPLACEMENT, and RANDOM SENTENCE conditions. Only the language instruction is modified.

A.6 MATCHED EVALUATION UNDER INSTRUCTION INTERVENTIONS

Across both LIBERO and RoboTwin 2.0, instruction interventions are evaluated under matched environmental conditions. Let s_0 denote an initial simulator state and ℓ the language instruction. For a

Table 3: Lexical pools used for the *Random Noun* and RANDOM SENTENCE interventions.

Pool	Members
LIBERO object vocabulary used for eligibility (45)	bowl, plate, ramekin, cup, mug, box, cabinet, drawer, basket, book, butter, cheese, chocolate, cream, sauce, can, bottle, pan, pot, lid, tray, rack, shelf, table, desk, chair, cloth, towel, bag, container, toy, block, cube, ball, hammer, brush, fork, spoon, knife, glass, jar, kettle, pitcher, bin, bucket.
Replacement and random-sentence nouns (99)	cat, dog, bird, fish, horse, lion, tiger, bear, wolf, rabbit, deer, eagle, shark, whale, frog, snake, turtle, rock, stone, tree, leaf, cloud, river, mountain, beach, forest, lake, desert, island, volcano, cave, valley, lamp, clock, mirror, pillow, blanket, carpet, curtain, vase, candle, picture, statue, phone, remote, cable, apple, banana, orange, grape, pizza, bread, rice, soup, cookie, cake, sandwich, salad, pasta, egg, milk, wrench, nail, screw, drill, rope, chain, key, lock, pipe, wire, tape, glue, paint, wheel, gear, shirt, hat, shoe, sock, coat, glove, belt, scarf, coin, stamp, ticket, card, map, flag, sign, label, button, switch, ring, medal, trophy, balloon, umbrella.
Random-sentence verbs (10)	observe, locate, inspect, examine, survey, approach, avoid, ignore, study, scan.
Random-sentence adjectives (10)	curious, ancient, shiny, purple, enormous, tiny, mysterious, hollow, rusty, invisible.

given evaluation instance, we keep s_0 fixed and replace only ℓ. Consequently, comparisons between ORIGINAL, NOUN REPLACEMENT, and RANDOM SENTENCE isolate the effect of language from variation in environment initialization.

Unless otherwise stated, success is measured using the native benchmark predicate and against the task specified by the original evaluation instance. Policies remain frozen throughout evaluation; no online adaptation or parameter update is performed.

B INSTRUCTION INTERVENTION DETAILS

This appendix provides the implementation details for the instruction interventions defined in Sec. 3.

B.1 NOUN REPLACEMENT

In-scene Target Replacement. For LIBERO-Object, we identify the manipulated target expression in the instruction. Each other manipulable object present in the same scene is used in turn as a replacement for the manipulated target, while the remaining instruction is left unchanged.

For example,

pick up the ketchup and place it in the basket → pick up the milk and place it in the basket.

Random Noun. For this variant, all eligible object expressions in the instruction are identified and replaced with randomly sampled nouns.

For LIBERO, eligible object expressions are identified by matching complete object names in the instruction against the 45-item vocabulary in Table 3. All matched expressions are replaced independently with nouns sampled uniformly from the 99-item replacement vocabulary.

For RoboTwin 2.0, eligible object expressions are resolved from the task-instruction template, `episode_info`, and the corresponding object-description metadata. This resolution includes episode-specific and multiword object descriptions. Every resolved eligible object expression is replaced, with each replacement noun sampled independently and uniformly from the same 99-item replacement vocabulary.

B.2 RANDOM SENTENCE GENERATION

For RANDOM SENTENCE, the original instruction is discarded. One of the ten templates in Table 4 is sampled uniformly. Each noun, verb, and adjective slot is filled independently and uniformly, with replacement, from the corresponding lexical pool in Table 3. No task-dependent, scene-dependent, semantic, or grammatical filtering is applied.

Table 4: Templates used to construct the RANDOM SENTENCE intervention.

No.	Template
1	`The {adj} {noun1} is resting next to the {noun2}.`
2	`A {noun1} and a {noun2} are placed on the {noun3}.`
3	`Please {verb} the {adj} {noun1} carefully.`
4	`There is a {noun1} near the {noun2} and a {noun3} on the floor.`
5	`The robot should {verb} the {noun1} without touching the {noun2}.`
6	`A {adj} {noun1} has been spotted beside the {noun2}.`
7	`Move the {noun1} towards the {adj} {noun2}.`
8	`The {noun1} is on top of the {noun2} which is next to the {noun3}.`
9	`Find the {adj} {noun1} and bring it to the {noun2}.`
10	`The {noun1} fell off the {noun2} and landed on the {noun3}.`

C SCENE–TARGET EVALUATION SETS

This appendix provides the complete scene–target sets used for the representation analyses described in Sec. 3. For each scene s, we define a six-element instructed-target set $\mathcal{T}_s$. The scene configuration and robot state are fixed across targets in $\mathcal{T}_s$, and only the instructed target changes.

C.1 LIBERO-OBJECT

All LIBERO-Object conditions use the instruction template

pick up the {*target*} *and place it in the basket.*

For each fixed scene s, the target slot is instantiated separately with each element of $\mathcal{T}_s$, yielding six scene–target instances. Tables C.1 show the scenes and corresponding target sets.

C.2 REAL-WORLD SCENES

The real-world scene–target set uses the same instruction template as LIBERO-Object,

pick up the {*target*} *and place it in the basket.*

All six scenes share the same six-element instructed-target set $\mathcal{T}_{\text{real}}$. For each fixed scene, the target slot is instantiated separately with each element of $\mathcal{T}_{\text{real}}$, while only the scene layout differs across scenes. The six scenes are shown in Table 6.

D LINEAR PROBE SETTINGS

We use linear probes to test whether the currently instructed target and the underlying scene identity remain explicitly accessible in the internal instruction representations of the policy. We conduct this analysis on $\pi_{0.5}$ and GR00T N1.7, comparing their base and fine-tuned checkpoints on the matched scene–target evaluation sets introduced in Sec. 3. Policy parameters remain frozen throughout all probing experiments.

D.1 PROBE TASKS AND EVALUATION SETS

We construct two separate classification tasks.

Target classification. The target probe predicts the identity of the object specified by the *current* instruction. Importantly, the label follows the modified instruction rather than the original task associated with the scene. For example, when

pick up the ketchup and place it in the basket

is changed to

Table 5: **LIBERO-Object scene–target sets.** Each scene s is associated with a six-element instructed-target set $\mathcal{T}_s$.

	Scene	Target set		Scene	Target set
1		$\mathcal{T}_1$ = { *alphabet soup*, *salad dressing*, *cream cheese*, *milk*, *tomato sauce*, *butter* }	2		$\mathcal{T}_2$ = { *cream cheese*, *alphabet soup*, *milk*, *tomato sauce*, *butter*, *orange juice* }
3		$\mathcal{T}_3$ = { *milk*, *cream cheese*, *tomato sauce*, *butter*, *orange juice*, *chocolate pudding* }	4		$\mathcal{T}_4$ = { *tomato sauce*, *milk*, *butter*, *orange juice*, *chocolate pudding*, *bbq sauce* }
5		$\mathcal{T}_5$ = { *butter*, *tomato sauce*, *orange juice*, *chocolate pudding*, *bbq sauce*, *ketchup* }	6		$\mathcal{T}_6$ = { *orange juice*, *butter*, *chocolate pudding*, *bbq sauce*, *ketchup*, *salad dressing* }
7		$\mathcal{T}_7$ = { *chocolate pudding*, *orange juice*, *bbq sauce*, *ketchup*, *salad dressing*, *alphabet soup* }	8		$\mathcal{T}_8$ = { *bbq sauce*, *chocolate pudding*, *ketchup*, *salad dressing*, *alphabet soup*, *cream cheese* }
9		$\mathcal{T}_9$ = { *ketchup*, *bbq sauce*, *salad dressing*, *alphabet soup*, *cream cheese*, *milk* }	10		$\mathcal{T}_{10}$ = { *salad dressing*, *ketchup*, *alphabet soup*, *cream cheese*, *milk*, *tomato sauce* }

pick up the milk and place it in the basket,

the target label changes from *ketchup* to *milk*.

Scene classification. The scene probe predicts the identity of the underlying visual scene. Because target replacement leaves the initial environment unchanged, all instruction variants constructed from the same initial scene share the same scene label.

We use the matched scene–target sets described in Sec. 3. LIBERO-Object contains ten scenes, each paired with six instructed targets, yielding

$$N_{\text{LIBERO}} = 10 \times 6 = 60$$

scene–target conditions. The real-world set contains six scenes, each paired with six instructed targets, yielding

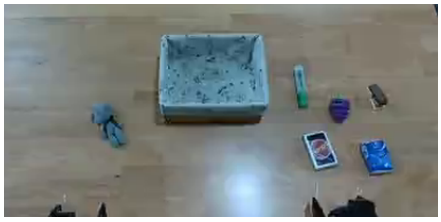
$$N_{\text{real}} = 6 \times 6 = 36.$$

Within each scene, the visual configuration and initial robot state remain fixed while only the instructed target changes.

D.2 Feature Extraction

For each model checkpoint, we run a forward pass for every scene–target condition and extract the hidden states associated with the instruction tokens at each analyzed transformer layer.

Table 6: **Real-world scene layouts.** All six scenes use the same instructed-target set $\mathcal{T}_{\text{real}}$ and the same instruction template.

1		2	
3		4	
5		6	

Let

$$H^{(\ell)} = \left[h_1^{(\ell)}, \ldots, h_{|T|}^{(\ell)}\right]$$

denote the hidden states at layer ℓ for the lexical instruction tokens indexed by T. Padding tokens and tokens outside the instruction span are excluded. We obtain one representation per condition and layer by mean-pooling over the instruction tokens:

$$\bar{h}^{(\ell)} = \frac{1}{|T|} \sum_{i \in T} h_i^{(\ell)}. \tag{5}$$

This produces a feature vector

$$\bar{h}^{(\ell)} \in \mathbb{R}^D$$

for every scene–target condition, where D is the hidden dimension of the corresponding model.

We perform feature-wise z-score standardization before fitting the probe. For feature dimension d,

$$\tilde{h}_{n,d}^{(\ell)} = \frac{\bar{h}_{n,d}^{(\ell)} - \mu_d^{(\ell)}}{\sigma_d^{(\ell)} + \epsilon}, \tag{6}$$

where $\mu_d^{(\ell)}$ and $\sigma_d^{(\ell)}$ denote the mean and standard deviation of feature d, respectively. The resulting standardized representations are used as inputs to the linear probes.

D.3 Label Construction

Target identity and scene identity are treated as two independent multiclass classification problems. For either task, a categorical label c_n is represented using a one-hot vector

$$y_n \in \{0,1\}^C,$$

where C is the number of classes for the corresponding prediction task and

$$y_{n,c} = \begin{cases} 1, & c = c_n, \\ 0, & \text{otherwise.} \end{cases}$$

For target classification, c_n denotes the object specified by the current instruction. For scene classification, c_n denotes the scene from which the condition was constructed. The two probe tasks therefore use the same representations but different label matrices.

D.4 Ridge-Regression Probe

For every analyzed layer, model checkpoint, environment, and prediction task, we train an independent linear probe. Let

$$X \in \mathbb{R}^{N \times D}$$

contain the standardized instruction representations and

$$Y \in \{0,1\}^{N \times C}$$

the corresponding one-hot labels.

We use squared-loss ridge regression:

$$\min_{W,b} \left\| XW + \mathbf{1}b^\top - Y \right\|_F^2 + \lambda \|W\|_F^2, \tag{7}$$

where

$$W \in \mathbb{R}^{D \times C}$$

is the linear readout, $b \in \mathbb{R}^C$ is the intercept, and λ controls L_2 regularization.

At inference time, the predicted class is obtained from the largest regression score:

$$\hat{c}_n = \arg\max_c \left(x_n^\top W + b \right)_c . \tag{8}$$

We use the same probe formulation for target and scene classification; only the labels differ.

D.5 Regularization Selection and Evaluation

The regularization coefficient λ is selected independently for each layer and classification task using 5-fold cross-validation on the training set. We search over logarithmically spaced values between

$$10^{-3} \quad \text{and} \quad 10.$$

After selecting λ, the ridge probe is fitted using the training samples and evaluated on held-out samples that are not used to fit the probe. Performance is reported as multiclass classification accuracy:

$$\text{Acc} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathbb{I}\left[\hat{c}_n = c_n\right]. \tag{9}$$

A separate probe is trained for every model, checkpoint (Base or Fine-tuned), environment (LIBERO-Object or real world), layer, and prediction target (Object or Scene). Policy parameters are never updated by the probing objective.

This setup measures *linear accessibility*: high probe accuracy shows that target or scene identity can be recovered through a linear readout of the representation. It does not establish that the decoded information is causally used by the policy or that the instructed target is correctly grounded in the visual observation.

E Layer-wise Action Lens Analysis

This section provides additional implementation details for the layer-wise action lens analysis in Sec. 5.2. We describe the counterfactual replay protocol, the construction of intermediate action estimates, the model-specific readouts, and the paired MSE used for comparison.

Model	Denoising steps	Layers / step	Evaluated action region	Instantiation of $\mathcal{R}_{m,s}$
$\pi_{0.5}$	10	18	10×7	Timestep-conditioned adaptive normalization followed by the native `action_out_proj`
GR00T N1.7	4	32	16×7	Native output normalization and timestep-conditioned modulation, followed by the output projection and LIBERO-specific embodiment mapping
FastWAM	10	30	32×7	Shared trained ActionDiT action head used in the native inference pathway
LingBot-VA	10	30	$16 \times 7^{*}$	Timestep-conditioned `norm_out` followed by `action_proj_out`

Table 7: **Model-specific action-lens readouts.** $\mathcal{R}_{m,s}$ is instantiated using the native output operations of each model. The evaluated action region contains only the valid LIBERO action coordinates used in the MSE computation. *For LingBot-VA, four context-fixed positions in the first conditioned chunk are excluded, yielding a 12×7 evaluated region for that chunk.

E.1 Action-Lens Evaluation Protocol

For each model, we use one successful closed-loop rollout on a LIBERO-Object task under the original instruction and record the policy inputs at every replanning event. At each recorded state, we rerun action generation under the original instruction and the interventions defined in Sec. 3. For the action-lens analysis, Noun Replacement includes both an out-of-scene noun and another visible object. Each intervention instruction is fixed across all replanning events in the rollout.

The counterfactual actions are not executed. At each replanning event, all non-linguistic inputs are held fixed across instruction conditions, including the observations, robot state, history or cache when applicable, denoising schedule, and generation noise. Thus, the analysis compares instruction conditions at matched policy states rather than through separate closed-loop rollouts.

E.2 Intermediate Action Readout

Let

$$\mathbf{h}_{m,r,q}^{(s,l)} \tag{10}$$

denote the action-token hidden state of model m at replanning event r, under instruction condition q, after layer l of denoising step s. Rather than training an additional probe, we reuse the model's frozen native action readout to decode each intermediate representation.

We denote this model-specific readout by $\mathcal{R}_{m,s}(\cdot)$ and define the corresponding intermediate action-space vector-field prediction as

$$\widehat{\mathbf{V}}_{m,r,q}^{(s,l)} = \mathcal{R}_{m,s}\left(\mathbf{h}_{m,r,q}^{(s,l)}\right). \tag{11}$$

Depending on the model architecture, $\mathcal{R}_{m,s}$ denotes the model-specific composition of native output operations used to map an intermediate action representation to the action-space vector field. The specific instantiation of $\mathcal{R}_{m,s}$ for each model is summarized in Table 7.

Let $\mathbf{X}_{m,r,q}^{(s)}$ denote the noisy action state before the update at denoising step s, and let $\Delta_{m,s}$ denote the signed integration coefficient used by the model's native inference procedure. We construct the intermediate action estimate as

$$\widehat{\mathbf{A}}_{m,r,q}^{(s,l)} = \mathcal{P}_m\left[\mathbf{X}_{m,r,q}^{(s)} + \Delta_{m,s}\widehat{\mathbf{V}}_{m,r,q}^{(s,l)}\right], \tag{12}$$

where $\mathcal{P}_m$ selects the valid LIBERO action horizon and the seven action coordinates used for evaluation.

The coefficient $\Delta_{m,s}$ follows each model's native action-update rule. All instruction conditions start from the same initial noisy action state,

$$\mathbf{X}^{(0)}_{m,r,q} = \mathbf{X}^{(0)}_{m,r,\text{orig}}. \tag{13}$$

For $s > 0$, their action states can differ because preceding denoising updates are conditioned on different instructions. We therefore compare intermediate readouts at matched denoising steps and network layers along each condition's own denoising trajectory.

E.3 MSE Computation

For each replanning event, instruction intervention, denoising step, and network layer, we compute the paired MSE between the intermediate action estimate and the corresponding estimate under the original instruction:

$$D^{(s,l)}_{m,r,q} = \frac{1}{H_{m,r}C} \sum_{h=1}^{H_{m,r}} \sum_{c=1}^{C} \left(\widehat{A}^{(s,l)}_{m,r,q,h,c} - \widehat{A}^{(s,l)}_{m,r,\text{orig},h,c} \right)^2, \tag{14}$$

where $H_{m,r}$ is the valid action horizon and $C = 7$ is the number of evaluated LIBERO action coordinates.

All distances are computed directly in each model's normalized action space; no denormalization to physical action units is applied. Only valid LIBERO action entries are included in the computation.

For Fig. 2, we average the paired distances over all replanning events in the selected successful rollout:

$$\overline{D}^{(s,l)}_{m,q} = \frac{1}{R_m} \sum_{r=1}^{R_m} D^{(s,l)}_{m,r,q}, \tag{15}$$

where R_m denotes the number of replanning events in the analyzed rollout.

F UMAP Representation Analysis

This section provides the implementation details for the UMAP analysis in Sec. 6.3, including representation extraction, normalization, and dimensionality-reduction settings.

F.1 Evaluation Set and Representation Extraction

The UMAP analysis uses the LIBERO-Object scene–target evaluation set introduced in Sec. 3. It contains ten scenes, each paired with six instructed targets: the original target and five alternative visible and reachable objects in the same scene, yielding

$$N = 10 \times 6 = 60 \tag{16}$$

scene–target conditions. Within each scene, the visual configuration and initial robot state remain fixed while only the instructed target changes.

For both the Base and fine-tuned checkpoints of $\pi_{0.5}$, we run a forward pass for every scene–target condition and extract the hidden states associated with the lexical instruction tokens at each analyzed transformer layer. Padding tokens and tokens outside the instruction span are excluded.

Let

$$H^{(\ell)}_n = \left[h^{(\ell)}_{n,1}, \ldots, h^{(\ell)}_{n,|T_n|} \right] \tag{17}$$

denote the hidden states at layer ℓ for the lexical instruction tokens indexed by T_n in condition n. We obtain one representation per condition and layer by mean-pooling over these tokens:

$$\bar{h}^{(\ell)}_n = \frac{1}{|T_n|} \sum_{i \in T_n} h^{(\ell)}_{n,i}. \tag{18}$$

This produces one feature vector $\bar{h}^{(\ell)}_n \in \mathbb{R}^D$ for each scene–target condition and analyzed layer, where D is the hidden dimension. The representation-extraction procedure follows the same instruction-token pooling protocol used in the linear probe analysis in App. D.

F.2 Feature Normalization

Before dimensionality reduction, we apply feature-wise z-score standardization across the 60 LIBERO-Object scene–target conditions. For each checkpoint and analyzed layer, let

$$X^{(\ell)} = \begin{bmatrix} (\bar{h}_1^{(\ell)})^\top \\ \vdots \\ (\bar{h}_N^{(\ell)})^\top \end{bmatrix} \in \mathbb{R}^{N \times D}. \tag{19}$$

For feature dimension d, the standardized representation is

$$\tilde{h}_{n,d}^{(\ell)} = \frac{\bar{h}_{n,d}^{(\ell)} - \mu_d^{(\ell)}}{\sigma_d^{(\ell)} + \epsilon}, \tag{20}$$

where $\mu_d^{(\ell)}$ and $\sigma_d^{(\ell)}$ denote the mean and standard deviation of feature d across the 60 conditions, respectively. Standardization is performed independently for each checkpoint and analyzed layer.

F.3 UMAP Dimensionality Reduction

For each checkpoint and analyzed layer, we independently apply UMAP to the standardized representation matrix $\tilde{X}^{(\ell)} \in \mathbb{R}^{60 \times D}$ and obtain a two-dimensional embedding:

$$Z^{(\ell)} = \mathrm{UMAP}\Big(\tilde{X}^{(\ell)}\Big) \in \mathbb{R}^{60 \times 2}. \tag{21}$$

The Base and fine-tuned checkpoints are embedded independently at each layer. We therefore interpret the visualization through the within-panel clustering structure rather than through absolute coordinates or distances across panels.

We use the same UMAP settings for all checkpoint–layer pairs: `n_neighbors`=10, `min_dist`=0.15, and `metric`=`euclidean`. All embeddings use two output dimensions.

Figure 6 reports representative layers 1, 6, 13, and 18. Scene identity is encoded by color and instructed-target identity by marker shape. These labels are used only for visualization and are not provided to UMAP during dimensionality reduction.

G Shared Semi-NMF and Variance Partitioning

Shared semi-NMF. We adapt the semi-NMF implementation in NMFLibrary (https://github.com/hiroyuki-kasai/NMFLibrary) to jointly factorize matched base and fine-tuned instruction representations. At each analyzed layer, we stack the two representation matrices:

$$X_{\mathrm{joint}} = [X_{\mathrm{base}}\ X_{\mathrm{FT}}] \in \mathbb{R}^{2N \times D}, \tag{22}$$

where $N = 60$ for LIBERO-Object. We fit $K = 12$ components by solving

$$\min_{A \geq 0, ; B} \frac{1}{2} \left| X_{\mathrm{joint}} - AB^\top \right|_F^2, \tag{23}$$

where $A \in \mathbb{R}_{\geq 0}^{2N \times K}$ contains nonnegative activations and $B \in \mathbb{R}^{D \times K}$ is an unconstrained signed basis. The first and last N rows of A correspond to the base and fine-tuned checkpoints, respectively. The shared basis therefore pairs components across checkpoints within each layer.

Variance partitioning. For each component and checkpoint, we regress activations across observed conditions on categorical scene labels, target labels, and both labels jointly. All three models include an intercept; the joint model contains additive main effects without interactions. Let their ordinary coefficients of determination be R_S^2, R_T^2, and R_{ST}^2. We partition activation variance as

$$U_{\mathrm{scene}} = R_{ST}^2 - R_T^2,\ U_{\mathrm{target}} = R_{ST}^2 - R_S^2,\ C_{\mathrm{shared}} = R_S^2 + R_T^2 - R_{ST}^2,\ R_{\mathrm{res}} = 1 - R_{ST}^2. \tag{24}$$

The unique terms measure the additional variance explained by each factor after controlling for the other. The shared term captures overlapping explanatory variation and can be negative under suppression; the residual is unexplained by the additive model.

We plot each component at $(U_{\text{target}}, U_{\text{scene}})$ and connect its base and fine-tuned positions. An upper-left shift indicates decreased target-unique variance and increased scene-unique variance. These quantities describe representational organization, not causal contributions to behavior.

H Detailed Literature Review

Recent advances in robot learning have expanded the ways perception is translated into action. This section organizes the literature into four complementary paradigms. **Vision–Action (VA) models** learn direct mappings from visual observations to motor commands, with an emphasis on precise sensorimotor control. **Vision–Language–Action (VLA)** models condition behavior on natural-language instructions, enabling a single policy to address a broader range of tasks. **World–Action Models (WAMs)** couple action generation with predictions of how the environment will evolve, providing a basis for anticipating action outcomes.

H.1 Vision–Action Models.

Vision–action (VA) models learn visuomotor policies that map visual observations and robot states to actions without requiring language instructions. Their development has centered on three related challenges: representing spatial information, generating coherent action sequences, and maintaining robust control during execution.

Diffusion-based policies have become a prominent approach to modeling multimodal action distributions. Diffusion Policy (Chi et al., 2025) established action-sequence denoising as an effective framework for visuomotor control. Subsequent work has strengthened its spatial representations: DP3 (Ze et al., 2024) conditions action generation on 3D point clouds, while iDP3 (Ze et al., 2025) extends this direction to humanoid manipulation learned from human demonstrations. Other methods refine how visual information enters the policy. H^3DP (Lu et al., 2026) introduces hierarchies across observations, visual features, and action generation; 3D Diffuser Actor (Ke et al., 2024) uses tokenized 3D scene representations; and Render and Diffuse (R&D) (Vosylius et al., 2024) aligns observations and candidate actions by rendering robot configurations in image space. Mamba Policy (Cao et al., 2025b) explores a more efficient sequence-modeling backbone, whereas MCDP (Cao et al., 2025a) combines diffusion policies trained on different visual modalities at inference time. GCP (Cao et al., 2026) further studies distribution-level policy composition across diffusion- and flow-based policies.

A second line of work addresses the structure and reliability of generated actions. DexDiffuser (Weng et al., 2024) targets dexterous grasp generation, while ALOHA Unleashed (Zhao et al., 2024) applies diffusion-based action generation to complex bimanual manipulation. ChainedDiffuser (Xian et al., 2023) connects end-effector keyposes to executable trajectories, and HDP (Ma et al., 2024) incorporates kinematic structure into policy learning. For generalization across objects and scenes, S^2-Diffusion (Yang et al., 2025) investigates the transfer from instance-level demonstrations to category-level skills, and C3DM (Saxena et al., 2023) uses constrained visual context to reduce the influence of distractors. Crossway Diffusion (Li et al., 2024) improves learned representations through an auxiliary self-supervised objective, while Equivariant Diffusion Policy (Wang et al., 2024) builds geometric symmetry into action generation.

Temporal consistency and responsiveness have motivated complementary advances. Streaming Diffusion Policy (Høeg et al., 2024) addresses continuous action generation during execution, and Bidirectional Decoding (BID) (Liu et al., 2025d) improves action-chunk execution through guided test-time sampling. Beyond diffusion, ACT (Zhao et al., 2023) predicts action chunks with a transformer-based policy, while Chunking Causal Transformer (Zhang et al., 2025) models dependencies across chunks. Act3D (Gervet et al., 2023) and Spatial Policy (Liu et al., 2025c) emphasize spatially grounded action prediction. Flow matching provides another route to generative control: ManiFlow (Yan et al., 2025) and VITA (Gao et al., 2026a) develop flow-based visuomotor policies, while Flow Matching Policy Gradients (McAllister et al., 2026) connects flow-based policies with reinforcement learning. Finally, DynaGuide (Du & Song, 2026) uses feedback from a dynamics model

to steer a diffusion policy at inference time, and Latent Policy Barrier (Sun & Song, 2026) uses learned latent boundaries to identify and respond to states outside the expert distribution.

H.2 Vision-Language-Action Models (VLA)

Vision-language-action (VLA) models extend visuomotor control by conditioning robot behavior on natural-language instructions. Their central challenge is to connect semantic information—such as the requested object, goal, or task—to actions that are executable in the observed scene. Existing approaches differ in where this connection is made: some learn language-conditioned policies directly, while others adapt pretrained vision-language models for action prediction or use language models to guide a separate control policy.

Early language-conditioned approaches explored several ways to ground goals in behavior. MDT (Reuss et al., 2024) learns a diffusion policy from language and visual goal specifications, while PlayFusion (Chen et al., 2023) learns skills from language-annotated play data. StructDiffusion (Liu et al., 2022) uses language and object-centric 3D observations to generate physically plausible rearrangement goals. Rather than changing the policy architecture, ROSIE (Yu et al., 2023) uses text-guided image generation to augment robot training data. These works establish complementary routes from linguistic goals to visuomotor behavior.

Scaling beyond individual tasks and platforms has motivated training on diverse robot data. RT-1 (Brohan et al., 2023) demonstrated large-scale instruction-conditioned robot learning, while RT-2 (Brohan et al., 2023) transferred knowledge from web-trained vision-language models into robotic action prediction. The Open X-Embodiment effort and RT-X models (O'Neill et al., 2024) extended this direction across datasets and embodiments; Octo (Team et al., 2024) likewise trains a generalist policy on heterogeneous robot demonstrations. Other methods address the scarcity or incompatibility of action data. LAPA (Ye et al., 2025) learns latent action representations from videos without requiring ground-truth robot actions during pretraining. RDT-1B (Liu et al., 2025b) uses a diffusion transformer and a unified action representation for bimanual manipulation, while RDT2 (Liu et al., 2026) investigates cross-embodiment generalization through larger-scale UMI data. TinyVLA (Wen et al., 2025a) instead emphasizes a compact vision-language backbone and a diffusion action decoder for efficient adaptation.

A major architectural question is how to preserve semantic reasoning while generating precise continuous actions. DiffusionVLA (Wen et al., 2025b) and HybridVLA (Liu et al., 2025a) combine autoregressive modeling with diffusion-based action generation. The π_0 family (Black et al., 2024; Intelligence et al., 2025; 2026) pairs a pretrained vision-language backbone with a flow-based action expert, with successive models extending its generalization and conditioning capabilities. ChatVLA (Zhou et al., 2025c) addresses the interaction between multimodal understanding and robot control through phased training and an expert-based architecture. Together, these models illustrate different ways to connect representations learned from vision and language to action generation.

Spatial reasoning and longer-horizon planning form a related direction. 3D-VLA (Zhen et al., 2024) connects 3D perception and language reasoning with generative predictions of future scenes, providing a bridge to world-action modeling. SpatialBot (Cai et al., 2025) strengthens depth-aware spatial understanding in vision-language models. At the planning level, HiP (Ajay et al., 2023) composes models specialized for language, visual prediction, and control, while Plan Diffuser (Sharan et al., 2024) grounds language-model plans through diffusion-based generation. ReKep (Huang et al., 2024) expresses tasks as spatial and temporal keypoint constraints, whereas VoxPoser (Huang et al., 2023) converts language-guided affordances into 3D value maps. These planning systems connect semantic goals to robot execution and lead into the agent-based approaches discussed later.

H.3 World-Action Models (WAM)

World-action models (WAMs) connect robot control with predictions of how the environment may change. This predictive signal can play different roles: a model may generate a visual plan before choosing actions, learn future observations and actions jointly, or use future prediction only as supervision during training. The distinction matters because forecasting a useful future representation does not necessarily require rendering future video at deployment. UniPi (Du et al., 2023) provides

an early example of planning through text-conditioned video generation, treating predicted visual trajectories as an intermediate representation for decision-making.

One approach separates visual prediction from action execution. Gen2Act (Bharadhwaj et al., 2024) generates videos of human behavior in novel scenes and uses them to guide robot manipulation. Vidar (Feng et al., 2025a) combines video diffusion pretraining with a masked inverse dynamics model that recovers actions, while CoT-VLA (Zhao et al., 2025) autoregressively predicts future images as visual goals before generating action sequences. Act2Goal (Zhou et al., 2025a) connects a goal-conditioned visual world model to control at multiple temporal scales. Related work uses video models to expand training data or support visual planning: DreamGen (Jang et al., 2025) produces synthetic robot trajectories from generated videos, and RoboDreamer (Zhou et al., 2024) factorizes video generation to compose visual plans from learned elements.

A second approach learns visual dynamics and actions within a shared model. GR-1 (Wu et al., 2024) predicts future images alongside robot actions, and GR-2 (Cheang et al., 2024) extends video pretraining to web-scale data before adapting to robot trajectories. WorldVLA (Cen et al., 2025b) and RynnVLA-002 (Cen et al., 2025a) further unify action prediction with world modeling. LingBot-VA (Li et al., 2026b) learns frame prediction and policy execution within an autoregressive diffusion framework. Unified World Models (UWM) (Zhu et al., 2025) couple video and action diffusion in a model that can be used for policy learning, visual prediction, and dynamics modeling. DreamVLA (Zhang et al., 2026) instead predicts compact world knowledge to support inverse dynamics and action generation.

Large video models offer another source of predictive representations. Cosmos Policy (Kim et al., 2026) adapts a pretrained video foundation model to visuomotor control through post-training, while DreamZero (Ye et al., 2026b) builds a joint video-action model on a pretrained video diffusion backbone. Genie Envisioner (Liao et al., 2026) brings policy learning, visual generation, and evaluation into a video-generative framework. DreamDojo (Gao et al., 2026b) pretrains an action-controllable world model on large-scale egocentric human video before adapting it to robot data. At the level of model design, DUST (Won et al., 2025) uses separate but interacting diffusion streams for visual futures and actions. Motus (Bi et al., 2026) and MotuBrain (Team et al., 2026) use multiple transformer streams to model understanding, video generation, and action within shared systems. LDA-1B (Lyu et al., 2026) scales joint learning of dynamics, policy, and visual forecasting across heterogeneous embodied data.

Explicit future-image generation, however, adds computational cost and may place demands on a model that differ from those of action prediction. FLARE (Zheng et al., 2025) introduces future-latent prediction as an implicit world-modeling objective for policy learning. Fast-WAM (Yuan et al., 2026) retains video co-training but omits future generation at test time, allowing a direct comparison between the benefits of predictive training and inference-time imagination. GigaWorld-Policy (Ye et al., 2026a) likewise centers action prediction while using visual dynamics to inform training. Across these methods, the central design choice is not simply whether to predict the future, but *what future information to learn, how to connect it to actions, and whether it must be generated during execution*.

H.4 Agentic Robotics

Agentic robotics studies how a robot can pursue a task through repeated observation, reasoning, action, and feedback. In these systems, a multimodal model may decompose a goal, select executable skills, inspect their outcomes, and revise its plan as the scene changes. The agent therefore coordinates perception and control over an entire task, rather than producing a single action from the current observation.

Recent work has made this coordination more explicit. Gemini Robotics 1.5 (Team et al., 2025b;a) combines embodied reasoning with a multi-embodiment action model. Reflective Planning (Feng et al., 2025b) uses imagined future states to reconsider skill choices, while RoboAgent (Xu et al., 2026) learns to chain basic capabilities for embodied task planning. MoMaStage (Li et al., 2026a) grounds planning in a skill-state graph, and Physical Agency (Galanti et al., 2026) places a closed-loop orchestrator above existing policies and robot skills. RoboClaw (Li et al., 2026c) extends agent coordination across data collection, policy learning, and task execution.

A newer line of work asks how much of the policy itself a general-purpose agent can provide. Agent as Policy (Jia et al., 2026) lets an agent inspect observations, write programs, issue robot commands, and adapt from physical feedback. GPT-Policy (Cheng et al., 2026) uses demonstrations and interaction history as context for verified robot tool calls. Evaluation of GPT-6 Astra examine direct action selection and its combination with a pretrained policy (Su et al., 2026); their results also expose the remaining difficulty of precise and dynamic control. Finally, GaP (Chen et al., 2026) uses coding agents and structured feedback to improve robot policies or executable skill graphs. Together, these studies shift attention toward the interfaces and feedback loops that connect general reasoning to reliable physical execution.