NOT ALL FEATURES ARE CREATED EQUAL: A MECHANISTIC STUDY OF VISION-LANGUAGE-ACTION MODELS

Bryce Grant[*] **Xijia Zhao**[*] **Peng Wang**[†]
Case Western Reserve University
{bag100, xxz1277, pxw206}@case.edu
https://cwru-aism.github.io/vla-interp-page/

ABSTRACT

Vision-Language-Action (VLA) models combine perception, language, and motor control in a single architecture, yet how they translate multimodal inputs into actions remains poorly understood. We apply activation injection, sparse autoencoders (SAEs), and linear probes to six models spanning 80M–7B parameters across 394,000+ rollout episodes on four benchmarks. The visual pathway dominates action generation across all architectures: injecting baseline activations into null-prompt episodes recovers near-identical behavior, while cross-task injection steers robots toward source-task positions (99.8% of X-VLA episodes align with the source trajectory), exposing spatially bound motor programs tied to scene coordinates rather than abstract task representations. Language sensitivity depends on task structure, not model design: when visual context uniquely specifies the task, language is ignored; when multiple goals share a scene, language becomes essential (X-VLA `libero_goal`: 94%→10% under wrong prompts vs. `libero_object`: 60–100% regardless). In all three multi-pathway architectures ($\pi_{0.5}$, SmolVLA, GR00T), expert pathways encode motor programs while VLM pathways encode goal semantics (2× greater behavioral displacement from expert injection), and subspace injection confirms these occupy separable activation subspaces. Per-token SAE processing is essential for action fidelity on most architectures, though mean-pooling improves fidelity on X-VLA. Contrastive identification recovers 82+ manipulation concepts, and causal ablation reveals sensitivity spanning 28–92% zero-effect rates independent of representation width. We release **Action Atlas** (https://action-atlas.com) for interactive exploration of VLA representations across all six models.

1 INTRODUCTION

Vision-Language-Action (VLA) models combine visual encoders, language backbones, and action decoders into end-to-end policies that generalize across objects and instructions without task-specific engineering (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024; Physical Intelligence, 2025; Physical Intelligence et al., 2025). Despite rapid adoption, a question remains: do these models actually follow language instructions, or do they replay visual-motor priors learned during fine-tuning?

This opacity presents practical challenges: when a VLA-controlled robot exhibits unexpected behavior, operators have no principled way to diagnose the failure. Current debugging is limited to behavioral observation, in contrast to classical robotics where kinematics and control models can be inspected and modified (Häon et al., 2025).

Sparse autoencoders (SAEs) can extract interpretable features from large language models (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2024), decomposing dense, polyse-

[*] Equal contribution.
[†] Corresponding author: pxw206@case.edu

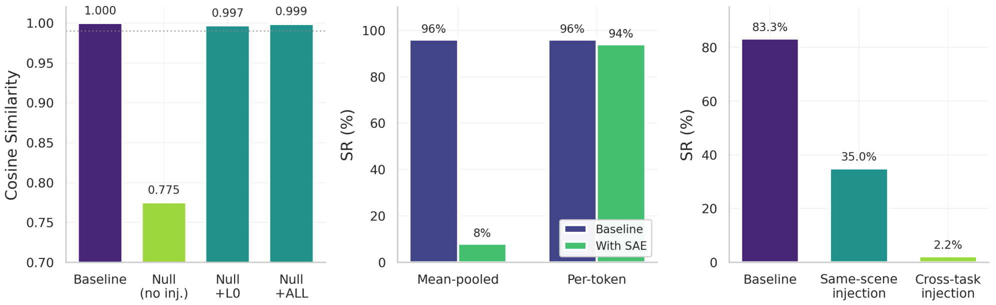


Figure 1: **Three core findings on $\pi_{0.5}$. Left:** Activation injection recovers baseline behavior from null-prompt episodes. Without injection, null prompts drop cosine similarity to 0.775; injecting a single layer (L0) recovers 0.997 and all layers recovers 0.999, demonstrating visual pathway dominance. **Middle:** Per-token SAE processing is essential. Mean-pooled SAE reconstruction destroys task success (96%→8%) despite high explained variance, while per-token processing preserves performance (96%→94%). **Right:** Cross-task injection fails destination tasks (83.3%→2.2%) and same-scene injection partially succeeds (35.0%), confirming spatially bound motor programs. These patterns replicate across all six models (Table 6).

mantic neural activations into sparse, monosemantic features corresponding to human-interpretable concepts. At scale, SAEs have revealed safety-relevant representations including deception and bias (Templeton et al., 2024), and have enabled activation steering for behavioral control without retraining (Turner et al., 2023; Rimsky et al., 2024). Recent work has applied SAEs to vision-language models (Pach et al., 2025; Joseph et al., 2025), but whether these methods extend to VLA behavior remains untested.

Applying mechanistic interpretability to VLAs presents challenges distinct from language models. VLAs process *heterogeneous token sequences* interleaving vision, language, and proprioception, and we find that mean-pooling activations across token positions destroys action-critical information, causing catastrophic task failure despite high reconstruction quality. *Causal validation* requires rollout-based evaluation: unlike LLMs where human judgment can assess output quality, VLA interpretability requires simulator or real-world rollouts to measure task success (Li et al., 2024b).

We present a mechanistic study across five VLAs ($\pi_{0.5}$ (Physical Intelligence, 2025), OpenVLA-OFT (Kim et al., 2025), X-VLA (Zheng et al., 2026), SmolVLA (Shukor et al., 2025), GR00T N1.5 (Bjorck et al., 2025)) and one language-free control (ACT (Zhao et al., 2023)), spanning 80M to 7B parameters, three action generation paradigms (flow matching, continuous regression, and CVAE), and four benchmarks (**394,000+ rollout episodes**). We establish four findings: (1) the visual pathway dominates behavior across all architectures; (2) language sensitivity is suite-dependent rather than architecture-dependent, with prompts ignored when visual context suffices but consequential in ambiguous scenes; (3) cross-task injection degrades destination success across all models (0% on five; -40pp on OFT) yet steers toward source positions, encoding spatially bound action sequences; and (4) multi-pathway architectures exhibit consistent specialization, with expert pathways encoding motor programs and VLM pathways encoding goals, a pattern that holds whether SAE processing is per-token or pooled.

Our contributions:

1. *Cross-architecture mechanistic analysis at scale*: the first systematic study spanning six architectures (80M–7B, four benchmarks, 394,000+ episodes). Visual pathway dominance, cross-task transfer failure, and suite-dependent language sensitivity replicate across all models.
2. *Pathway specialization*: consistent functional dissociations in $\pi_{0.5}$, SmolVLA, and GR00T N1.5, where expert pathways cause 2× greater behavioral displacement than VLM pathways.
3. *SAE-based causal analysis*: per-token processing is required for action fidelity (mean-pooling destroys behavior on most architectures), concept ablation across 15,096+ pairs

Model	Params	Layers	Dim	Action Gen.	Pathway	Bench.
$\pi_{0.5}$ (Physical Intelligence, 2025)	3B	18	1024	Flow (50 steps)	Dual (PG + Exp.)	LIBERO
OFT (Kim et al., 2025)	7B	32	4096	Cont. L1 regr.	Single (Llama-2)	LIBERO
X-VLA (Zheng et al., 2026)	$\sim$1B	24	1024	Flow matching	Single (Flor.-2)	LIB., SimpE.
SmolVLA (Shukor et al., 2025)	450M	32	960/480	Flow matching	Dual (VLM + Exp.)	LIB., MW
GR00T (Bjorck et al., 2025)	3B	32	varies	Diff./flow	Triple (D+E+V)	LIBERO
ACT (Zhao et al., 2023)	80M	–	–	CVAE	Enc.-Dec.	ALOHA

Table 1: **Architectures under study.** Five VLAs and ACT (language-free control). PG = PaliGemma, Exp. = Expert, Flor. = Florence, D+E+V = DiT + Eagle + VL-SA, LIB. = LIBERO, MW = MetaWorld, SimpE. = SimplerEnv. Full architecture details in Appendix K.1.

reveals causal sensitivity spanning 28–92% zero-effect rates independent of representation width, and 82+ manipulation concepts are identifiable via contrastive selection.

4. *Action Atlas*: open-source platform (https://action-atlas.com) for interactive exploration of VLA representations across all six models.

2 Related Work

VLA models extend vision-language pretraining to robotic control. RT-2 (Zitkovich et al., 2023) demonstrated that VLMs can generate tokenized robot actions; OpenVLA-OFT (Kim et al., 2025) replaced discrete tokenization with continuous L1 regression, achieving 97.1% LIBERO success. $\pi_{0.5}$ (Physical Intelligence, 2025) introduced flow matching with a dedicated action expert. The six models we study (Table 1) span this design space. Independent robustness evaluations (Fei et al., 2025; Zhou et al., 2025) reveal that VLAs collapse under perturbation (97% → 0% under 0.2-unit position shifts), motivating mechanistic investigation.

Mechanistic interpretability (Olah et al., 2020) has progressed through SAEs (Bricken et al., 2023; Cunningham et al., 2023; Templeton et al., 2024) and activation steering (Turner et al., 2023; Rimsky et al., 2024). For VLAs, Häon et al. (Häon et al., 2025) demonstrated SAE steering on π_0 and OpenVLA, Molinari et al. (Molinari et al., 2025) probed for emergent world models, and Khan et al. (Khan et al., 2025) used SAEs to isolate interpretable steering directions in Magma. We extend these single-architecture studies to cross-architecture validation across six models and show that action tokenization constrains SAE applicability. Extended related work appears in Appendix A.

3 Method

Our methodology combines four techniques (activation injection, counterfactual prompting, SAEs, and linear probes) applied uniformly across all six models. Figure 2 illustrates the overall pipeline.

3.1 VLA Architectures Under Study

We study six architectures spanning nearly two orders of magnitude in parameter count (80M–7B) and four action generation paradigms (flow matching (Lipman et al., 2023), continuous regression, diffusion, and CVAE decoding). Table 1 summarizes the architectural diversity.

3.2 Activation Injection

Our primary technique for establishing causal relationships is *activation injection*, an extension of activation patching (Meng et al., 2022) to full rollout episodes: replacing activations from one episode with those from another during inference. Given source episode A (correct prompt, successful rollout) and target episode B (alternative condition), we record layer activations $\{\mathbf{H}^{A,(\ell)}\}$ during episode A, then replace $\mathbf{H}^{B,(\ell)}$ with $\mathbf{H}^{A,(\ell)}$ at specified layers during episode B.

We test four conditions. In **null injection**, the source episode uses the correct prompt while the target uses an empty string. In **same-scene injection**, both episodes share the same visual scene but target different objects. In **cross-task injection**, source and target occupy entirely different visual

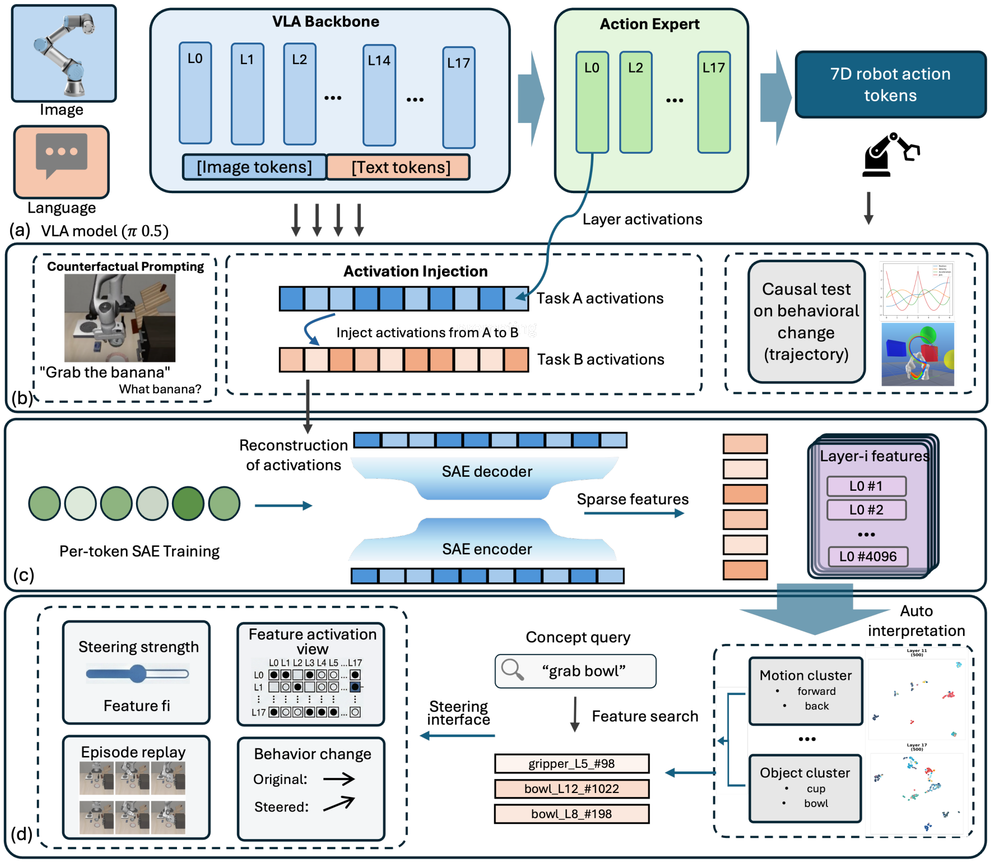


Figure 2: **Methodology overview.** Top: activations are recorded from VLA backbone and action expert layers during rollout episodes, then replayed under counterfactual conditions (null prompts, cross-task scenes) to establish causal relationships via behavioral change. Middle: per-token SAEs decompose layer activations into sparse features. Bottom: features are clustered, searched, and causally validated through ablation and steering experiments, with results visualized in Action Atlas.

scenes. In **cross-seed injection**, both episodes perform the same task under different random seeds. For multi-pathway models ($\pi_{0.5}$, SmolVLA, GR00T), we also inject into individual pathways to isolate their contributions.

3.3 COUNTERFACTUAL PROMPTING

We systematically vary text prompts to measure language sensitivity. Each episode is evaluated under one of six conditions: the baseline correct prompt, a null prompt (empty string), a negation prompt ("Don't pick up X"), a motor command ("Move slowly"), an object swap (replacing the target object name), and a temporal switch (changing the prompt mid-episode). For SmolVLA on MetaWorld, we also test counterfactual prompts across four difficulty levels (easy, medium, hard, very hard).

3.4 SPARSE AUTOENCODERS FOR VLAS

SAEs decompose dense neural activations into sparse, interpretable features. We train SAEs on action-relevant activations with TopK sparsity (Gao et al., 2024) ($k = 64$ active features) and expansion factor $m = 4d$ or $m = 8d$.

Per-Token Processing. VLA activations must be processed per-token. Mean-pooling across action tokens destroys heterogeneous temporal structure (approach, manipulation, and terminal phases encode distinct information), causing task failure despite high reconstruction quality ($R^2 > 0.95$). However, the relationship between pooling strategy and rollout fidelity is non-trivially architecture-dependent: in X-VLA, mean-pooled SAEs achieve *better* rollout fidelity than per-token despite lower training explained variance, while in GR00T, mean-pooling boosts VL-SA layer quality from 83–89% to 99% EV.

Feature Identification. We identify concept-specific features using frequency-weighted contrastive selection: $\text{score}_f = d_f \times \text{freq}_f$, where d_f is Cohen's d (Cohen, 1988) measuring activation difference between concept-present and concept-absent tasks, and freq_f is the fraction of samples where feature f appears in the active top-k.

Scale. Across all six models, we train **424 SAEs**: 96 for X-VLA (24 layers × 2 pooling strategies × 2 environments), 192 for SmolVLA (32 layers × 2 components × $\{$2 LIBERO pooling + Meta-World$\}$), 68 for GR00T (32 layers × 2 pooling + 4 VL-SA k128), and the original SAEs for $\pi_{0.5}$ (36) and OFT (32). These collectively identify **82+ unique manipulation concepts** across motion, object, and spatial categories.

3.5 Linear Probes for Action Prediction

Linear probes (Alain & Bengio, 2017) test whether action information is linearly decodable from intermediate representations. We train ridge regression probes for each action dimension and apply causality tests by projecting out the probe direction to verify whether predictive information is removed.

3.6 Metrics

We evaluate three primary metrics. **Action Cosine Similarity** measures behavioral alignment between episodes. **Task Success** is a binary indicator determined by the environment's built-in success criteria. **Override Rate** quantifies how often the robot follows injected behavior rather than the text prompt. All reported confidence intervals are 95% Wilson score intervals; ANOVA effect sizes are reported as η^2.

4 Experiments

We evaluate our methodology across five VLAs and one language-free control: $\pi_{0.5}$ (Physical Intelligence, 2025) (3B, flow-matching), OpenVLA-OFT (Kim et al., 2025) (7B, continuous L1 regression), X-VLA (Zheng et al., 2026) (1B, soft-prompted flow-matching), SmolVLA (Shukor et al., 2025) (450M, interleaved VLM-expert), GR00T N1.5 (Bjorck et al., 2025) (3B, DiT-Eagle-VL-SA hybrid), and ACT (Zhao et al., 2023) (80M, CVAE encoder-decoder, vision-only). Across all models, we collect **394,000+ rollout episodes** spanning 12 experiment types, 4 benchmarks, and up to 50 tasks per environment. Experiments were conducted on an 8×A100-SXM4-80GB cluster, an RTX 5090, and two RTX 4090s. Our experiments address five questions: (1) Does the visual pathway strongly influence behavior across architectures? (2) Do fine-tuned VLAs follow language instructions (for the five language-capable models)? (3) Does pathway specialization generalize across multi-component architectures? (4) How do SAE properties vary across architectures? (5) Do these phenomena hold across benchmarks and embodiments?

4.1 Experimental Setup

Benchmarks and Scale. We evaluate on four benchmarks: **LIBERO** (Liu et al., 2023) (4 suites, 40 tasks), **MetaWorld** (Yu et al., 2020) (50 tasks, 4 difficulty levels), **SimplerEnv** (Li et al., 2024b) (10 tasks, 2 embodiments), and **ALOHA** (Zhao et al., 2023) (2 bimanual tasks). Table 2 summarizes scale.

4.2 Visual Pathway Influence

We test the relative influence of the visual pathway versus language instructions across all six models.

Model	**Episodes**	**SAEs**	**Concepts**
$\pi_{0.5}$	65,000+	36	43
OpenVLA-OFT	70,700+	32	45
X-VLA	50,000+	96	82
SmolVLA	42,000+	192	45
GR00T N1.5	164,700+	68	36
ACT	1,870	–	–
Total	$>$**394,000**	$>$**424**	$>$**82**

Table 2: Experimental scale across six models. Episode counts aggregate across applicable experiment types (baselines, counterfactual prompting, cross-task injection, vision perturbation, grid ablation, SAE validation, concept ablation, concept steering, temporal ablation, fraction-to-failure); not all models undergo every type. SAE counts include per-token, mean-pooled, and k-sweep variants.

On $\pi_{0.5}$, supplying a null prompt while injecting baseline PaliGemma activations recovers near-identical behavior: cosine similarity between injected and baseline actions reaches 0.999, and task success recovers to 73–77% despite the absence of language (Table 3). Injecting only layer 0 achieves comparable results (0.997); task-relevant information is already encoded in the first transformer layer. On OpenVLA-OFT, null injection recovers only 14–15% success across all four LIBERO suites (n=120 per layer), a catastrophic drop from 90–100% baselines and far lower recovery than $\pi_{0.5}$'s 73–77%.

Model	**Condition**	**Goal**	**Object**	**Spatial**	**10/Long**$^{\diamond}$
$\pi_{0.5}$	Baseline	91.9%	–	77.3%	62.5%
	Null, no injection	0%	–	–	–
	Same-scene inject ALL	91.0%	–	–	75.2%
OFT	Baseline	100%	100%	90%	90%
	Null + zero any layer	14.1%	14.6%	14.4%	13.5%
X-VLA	Baseline	96.7%	100%	90.0%	100%
	Zero any single layer	0%	0%	0%	0%
SmolVLA	Baseline	75.0%	78.5%	67.5%	40.7%
	Zero expert layer	60–83%	60–83%	47–77%	0–33%
GR00T	Baseline	93.3%	93.3%	–	83.3%$^{\diamond}$
	Null source (ALL DiT)	3.3%	3.3%	–	3.3%$^{\diamond}$
	Zero any DiT layer	0%	0%	–	0%$^{\diamond}$
ACT	Baseline	100% (2 ALOHA tasks)			
	Mask grid (2,2)	10%			

Table 3: **Visual pathway influence across all six architectures, per suite.** *Baseline*: correct prompt, no intervention. *Same-scene inject*: inject baseline activations into same-task episode. *Null source*: null prompt with activation injection. *Zero*: zeroing all activations at a layer. $^{\diamond}$GR00T uses `libero_long` (no `libero_10`). $\pi_{0.5}$ baselines from cross-task experiment (n=138/23/24 pairs); same-scene injection: goal n=390, 10 n=330. OFT: n=120 per layer per suite. GR00T null injection: n=30 per suite (10 tasks × 3). SmolVLA: expert-layer ranges reflect L0 (catastrophic) through L31 (near-baseline).

Within a shared scene, injection overrides target selection. Running with prompt A while injecting activations from prompt B yields 93.3% behavioral override: the visual pathway overrides language conditioning. Same-scene injection on OpenVLA-OFT *hurts* performance (-17pp to -57pp) because injection steers toward the source behavior at the cost of the original task.

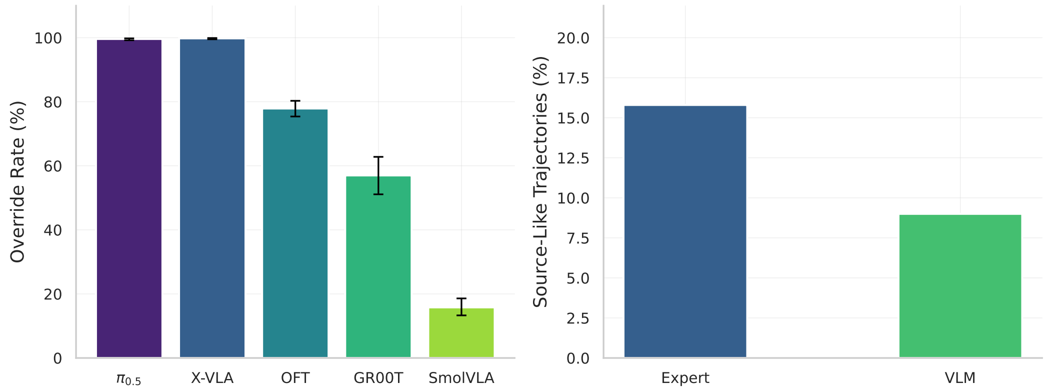


Figure 3: **Cross-task displacement override rates.** Left: override rate across five models. $\pi_{0.5}$ (99.6%, n=1,968) and X-VLA (99.8%, n=3,150) show near-complete source behavior transfer; OFT 77.9% (n=1,079); GR00T 57.0% (n=270, suite-dependent: `goal` 85.6%, `long` 33.3%). Error bars: 95% Wilson CIs. Right: SmolVLA pathway displacement (15.8% expert vs. 9.0% VLM, 732 pairs).

On X-VLA, every single layer is critical: zeroing any one of the 24 layers causes 0% task success on both LIBERO and SimplerEnv-WidowX. All 24 layers contribute essential information to action generation.

4.3 Cross-Task Transfer and Displacement Analysis

Cross-task injection causes catastrophic task failure across all six models, but trajectory analysis reveals that injected activations *do* steer behavior toward source-task positions. Table 4 summarizes the cross-task results.

Model	**Pairs**	**Success**	**Displacement**
$\pi_{0.5}$	1,968	2.6%	cos(traj, src) > cos(traj, dst): 99.6%
OFT	1,079	∼50%[†]	cos(traj, src) > cos(traj, dst): 77.9%
X-VLA	3,150	0%	cos→src > dst (99.8%)
SmolVLA	732	0%	Expert: 15.8% source-like
GR00T	270	0%	cos→src > dst (57.0%)

Table 4: Cross-task injection across five VLA models. Destination task success collapses universally, but displacement analysis reveals architecture-dependent behavioral steering. [†]OFT: suite-dependent; `libero_goal` drops 40pp from baseline (∼50%); other suites maintain near-baseline rates due to OFT's wide 4096-dim representations.

Displacement analysis (Figure 3) resolves an apparent contradiction: cross-task injection "fails" in terms of task success but "succeeds" in steering behavior. $\pi_{0.5}$ (99.6%), X-VLA (99.8%), and OFT (77.9%) show strong source-dominant trajectories; GR00T shows moderate override (57.0%), with a clear complexity gradient: `goal` (108-step mean baseline) at 85.6%, `object` (136 steps) at 52.2%, `long` (303 steps) at 33.3%; SmolVLA's expert pathway causes ∼2× greater displacement than its VLM pathway (15.8% vs. 9.0%). The robot reaches toward where source objects *would have been*, executing what we term *spatially grounded motor programs*: action sequences bound to specific scene coordinates rather than abstract task representations (Fei et al., 2025; Zhou et al., 2025). This contrasts with the object-centric abstractions typically assumed in task and motion planning: behavior cloning binds motor programs to absolute workspace coordinates rather than relational object representations.

4.4 Language Sensitivity

Counterfactual prompting across 3,396 episodes on $\pi_{0.5}$, 900 on OFT (3 conditions × 4 suites), 4,800 on X-VLA (24 conditions × 4 suites), 1,500 on SmolVLA (MetaWorld, 4 difficulty levels), and 630 on GR00T reveals that language sensitivity depends on task structure, not architecture. On $\pi_{0.5}$, ANOVA across prompt categories yields $F(4, 3391) = 1.23$, $p = 0.247$, $\eta^2 = 0.012$ (negligible effect size). The model achieves near-baseline performance even with null prompts: given an empty string, it executes a coherent manipulation sequence determined entirely by the visual scene.

Model / Suite	Baseline	Null	Wrong Obj.
$\pi_{0.5}$ / object	77.4%	77.0%	74.2%
$\pi_{0.5}$ / goal	83.3%	80.0%	76.7%
OFT / object	100%	100%	100%
OFT / goal	100%	10%	10%
OFT / spatial	90%	70%	60%
X-VLA / object	100%	60%	60–90%
X-VLA / goal	94%	10%	4–10%
X-VLA / spatial	98%	48%	44–58%
SmolVLA / MW easy	85%	82%	–
SmolVLA / MW hard	62%	41%	–
GR00T / object	93%	70%	50%
GR00T / goal	97%	0%	0%
GR00T / long	83%	67%	47%

Table 5: Counterfactual prompting across models (n=900 OFT, n=3,396 $\pi_{0.5}$, n=4,800 X-VLA, n=630 GR00T). $\pi_{0.5}$ ignores all prompt variations (p=0.247, ANOVA). OFT mirrors X-VLA's suite-dependent pattern: `object` is immune (100%) but `goal` collapses to 10% under generic/wrong prompts. SmolVLA shows difficulty-dependent sensitivity on MetaWorld. GR00T shows strong suite-dependent sensitivity: `libero_goal` collapses from 97% to 0% under null and wrong-task prompts, while `libero_object` retains 50–77% across conditions.

Despite behavioral invariance on $\pi_{0.5}$, linear classifiers trained on layer 17 activations predict prompt category with 99.3% accuracy: prompts are encoded but not used. Table 5 shows the suite-dependent pattern replicates across architectures: `libero_object` is prompt-immune on OFT (100%) and near-immune on X-VLA (60–100%), while `libero_goal` collapses to 0–10% under wrong prompts on OFT, X-VLA, and GR00T. SmolVLA mirrors this on MetaWorld: easy tasks are language-insensitive while harder tasks show greater sensitivity. The common factor is whether visual context alone identifies the target, not model design.

4.5 Pathway Specialization Across Architectures

Three of our six models feature distinct internal pathways, enabling analysis of functional specialization that generalizes across architectural designs.

$\pi_{0.5}$ (Dual: PaliGemma + Expert). Injecting expert activations from a mismatched task produces active wrong behavior (reaching toward incorrect locations, mean episode length 231–337 steps). Injecting PaliGemma activations produces passive stalling (running to the full 520-step limit). This dissociation establishes that the expert encodes motor programs ("how") while PaliGemma encodes goal semantics ("what"). Probes confirm: expert activations achieve $R^2 = 0.45$ for state prediction and AUC $= 0.93$ for success prediction; PaliGemma achieves $R^2 \approx 0$ but 76.4% goal classification accuracy.

SmolVLA (Dual: VLM + Expert, Interleaved). Grid ablation across all 32 layers of both pathways (VLM and expert) reveals suite-dependent sensitivity. Expert layer 0 is critical: zeroing it drops success to 0% on `libero_10` (baseline 41%) and 47% on `libero_spatial` (baseline 68%), while later expert layers maintain near-baseline performance (60–83% on `libero_goal` and `libero_object`). On MetaWorld, where both pathway types have complete 32-layer coverage, VLM early-layer zeroing is comparably destructive (mean 5–8pp worse than expert zeroing across difficulty levels). However, cross-task injection through expert layers causes $\sim 2\times$ greater behavioral displacement than VLM layers (15.8% vs. 9.0% source-like behavior across 732 Meta-World pairs), indicating that expert layers encode motor programs while VLM provides task context. Vision perturbation confirms spatial encoding: the model tolerates color jitter (-5pp) but fails under crops (-85pp) and flips (-92pp). Oracle probes confirm specialization: expert activations capture 58% of ground-truth state information at horizon 10, while VLM captures only 13%.

GR00T N1.5 (12 Eagle LM + 4 VL-SA + 16 DiT = 32 layers). Across all 96 layer-suite combinations (32 layers × 3 LIBERO suites, 164,700+ episodes), the three layer types serve distinct roles. DiT layers (98–99% SAE EV) are most ablation-sensitive (40–80% success drop); Eagle LM layers show moderate sensitivity; VL-SA layers are the most resilient despite lower per-token SAE quality (83–89% EV; mean-pooling boosts to 99%). Probes achieve 100% task identification and 96.4% success prediction across all 32 layers (Table 24). Expert pathways consistently encode motor programs while VLM pathways encode goal semantics, a specialization that holds across sequential ($\pi_{0.5}$), interleaved (SmolVLA), and triple-component (GR00T) designs. This specialization enables runtime failure diagnosis: expert-pathway injection produces active misdirection while VLM-pathway injection produces passive stalling, so monitoring pathway-specific activation norms distinguishes motor errors from goal errors.

4.6 SAE Analysis Across Architectures

Per-Token vs. Mean-Pooled. $\pi_{0.5}$ and OFT require per-token processing (mean-pooling: 0.4% vs. 70% success on $\pi_{0.5}$). X-VLA is the exception: mean-pooled SAEs achieve 94–100% rollout fidelity vs. 92–98% for per-token, despite 20–39% dead features. Florence-2's soft-prompted action tokens produce more uniform representations across positions; the dead features filter positional noise rather than discarding information. GR00T VL-SA layers similarly benefit from pooling (83% → 99% EV). Contrastive concept identification recovers **82+ manipulation concepts** across all models (Appendix F.1).

Causal Profiles Across Five Architectures. Concept ablation across 15,096+ concept-task pairs reveals causal sensitivity that does not follow representation width. SmolVLA (480-dim) is most sensitive (28% zero effect, 6.3% destruction); $\pi_{0.5}$ (1024-dim) is bimodal (54% zero, 14% destruction); GR00T shows pathway-specific sensitivity (DiT: 56% zero vs. Eagle: 73% zero); OFT (4096-dim) and X-VLA (1024-dim) are resilient (92% and 82% zero effect). Kill-switch features (single concepts whose ablation causes $>$50pp drops) concentrate at early layers: SmolVLA L0–L1 account for all expert kill-switches (13% destruction each, vs. $<$2% at L2+); $\pi_{0.5}$ peaks at L8 (40% destruction); GR00T DiT L0 is most destructive (21%) while Eagle and VL-SA layers show $<$3%. Across models, 70% of kill-switches encode object concepts (bowl, cabinet, mug) rather than motion primitives, and their scope scales inversely with width: $\pi_{0.5}$ kill-switches affect 21 tasks on average vs. 5 on OFT. This layer- and concept-type specificity indicates that kill-switches correspond to early-stage object binding rather than late-stage motor execution.

5 Conclusion

VLA representations are rich (82+ concepts, specialized pathways) yet brittle: cross-task injection steers toward source positions without task success, and language use is suite-dependent. Pathway specialization enables failure diagnosis: probing expert vs. VLM activations distinguishes motor errors from goal misidentification. Across 394,000+ episodes and six architectures, the gap between representation richness and behavioral robustness motivates new alignment methods. Action Atlas: https://action-atlas.com.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei

Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Pierre Sermanet, Nicolas Tomas, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, and Andy Zeng. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. doi: 10.48550/arXiv.2204.01691.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (Workshop)*, 2017. URL https://arxiv.org/abs/1610.01644.

ALOHA 2 Team, Jorge Aldaco, Travis Armstrong, Chelsea Finn, Pete Florence, Jonathan Tompson, and Tony Z. Zhao. ALOHA 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024. URL https://arxiv.org/abs/2405.02292.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. URL https://arxiv.org/abs/1506.03099.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. doi: 10.48550/arXiv.2410.24164. URL https://arxiv.org/abs/2410.24164.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, Hao Niu, Wenxuan Ou, Wanli Peng, Zeyu Ren, Haixin Shi, Jiawen Tian, Hongtao Wu, Xin Xiao, Yuyang Xiao, Jiafeng Xu, and Yichu Yang. GR-3 technical report. *arXiv preprint arXiv:2507.15493*, 2025. URL https://arxiv.org/abs/2507.15493.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023. URL https://arxiv.org/abs/2309.08600.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2409.14674.

Shaoqi Dong, Chaoyou Fu, Haihan Gao, Yi-Fan Zhang, Chi Yan, Chu Wu, Xiaoyu Liu, Yunhang Shen, Jing Huo, Deqiang Jiang, Haoyu Cao, Yang Gao, Xing Sun, Ran He, and Caifeng Shan. VITA-VLA: Efficiently teaching vision-language models to act via action expert distillation. *arXiv preprint arXiv:2510.09607*, 2025. URL https://arxiv.org/abs/2510.09607.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. doi: 10.48550/arXiv.2303.03378.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024. URL https://arxiv.org/abs/2410.00371.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. URL https://arxiv.org/abs/2510.13626.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024. doi: 10.48550/arXiv.2406.04093. URL https://arxiv.org/abs/2406.04093.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. CAST: Counterfactual labels improve instruction following in vision-language-action models. *arXiv preprint arXiv:2508.13446*, 2025. URL https://arxiv.org/abs/2508.13446.

Michal Golovanevsky, William Rudman, Michael Lepori, Amir Bar, Ritambhara Singh, and Carsten Eickhoff. Pixels versus priors: Controlling knowledge priors in vision-language models through visual counterfacts. *arXiv preprint arXiv:2505.17127*, 2025. URL https://arxiv.org/abs/2505.17127.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2743–2762. PMLR, 2025. URL https://arxiv.org/abs/2509.00328.

Sonia Joseph, Praneet Suresh, Ethan Goldfarb, Lorenz Hufe, Yossi Gandelsman, Robert Graham, Danilo Bzdok, Wojciech Samek, and Blake Aaron Richards. Steering clip's vision transformer with sparse autoencoders, 2025. URL https://arxiv.org/abs/2504.08729.

Hamza Khan et al. Controlling vision-language-action policies through sparse latent directions. In *NeurIPS 2025 Workshop on Mechanistic Interpretability*, 2025. URL https://openreview.net/pdf?id=wtf3ww1EOL.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://droid-dataset.github.io/.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. doi: 10.48550/arXiv.2406.09246. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2502.19645.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA training via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.09674.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. CoA-VLA: Improving vision-language-action models via visual-textual chain-of-affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2412.20451.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a. URL https://arxiv.org/abs/2411.19650.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Andrea Thomaz, Quan Vuong, Tianhe Yu, Chelsea Finn, and Sergey Levine. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, 2024b. URL https://simpler-env.github.io/. SIMPLER: simulation-based evaluation for real-world policies.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023. URL https://arxiv.org/abs/2209.07753.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2023. URL https://arxiv.org/abs/2210.02747. Alias for lipman2023flow.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://arxiv.org/abs/2306.03310.

Zhenyang Liu, Yongchong Gu, Sixiao Zheng, Yanwei Fu, Xiangyang Xue, and Yu-Gang Jiang. TriVLA: A triple-system-based unified vision-language-action model with episodic world modeling for general robot control. *arXiv preprint arXiv:2507.01424*, 2025. URL https://arxiv.org/abs/2507.01424.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://arxiv.org/abs/2202.05262.

Marco Molinari et al. Emergent world representations in OpenVLA. *arXiv preprint arXiv:2509.24559*, 2025. doi: 10.48550/arXiv.2509.24559. URL https://arxiv.org/abs/2509.24559.

Aaron Mueller. Missed causes and ambiguous effects: Counterfactuals pose challenges for interpreting neural networks. *arXiv preprint arXiv:2407.04690*, 2024. URL https://arxiv.org/abs/2407.04690.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parber, Austin Lo, Abhishek Jain, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://robocasa.ai/.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. doi: 10.1016/S0042-6989(97)00169-7.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2024. URL https://arxiv.org/abs/2310.08864.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://arxiv.org/abs/2504.02821.

Karl Pertsch, Kyle Walke, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL https://arxiv.org/abs/2501.09747.

Nicholas Pfaff, Haochen Dai, Yutong Qiu, and Pulkit Agrawal. SceneSmith: Agentic generation of simulation-ready indoor scenes. *arXiv preprint arXiv:2602.09153*, 2026. URL https://arxiv.org/abs/2602.09153.

Physical Intelligence. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Chelsea Finn, Sergey Levine, et al. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025. URL https://arxiv.org/abs/2511.14759.

Joris Postmus and Steven Abreu. Steering large language models using conceptors: Improving addition-based activation engineering. *arXiv preprint arXiv:2410.16314*, 2024. URL https://arxiv.org/abs/2410.16314. Workshop on Foundation Model Interventions at NeurIPS 2024.

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

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-flow models. In *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3736–3761. PMLR, 2025. URL https://arxiv.org/abs/2509.04996.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.828/.

Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Som Sagar, Jiafei Duan, Sreevishakh Vasudevan, Yifan Zhou, Heni Ben Amor, Dieter Fox, and Ransalu Senanayake. RoboFail: Analyzing failures in robot learning policies. *arXiv preprint arXiv:2412.02818*, 2024. URL https://arxiv.org/abs/2412.02818.

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

Anushka Sivakumar, Andrew Zhang, Zaber Ibn Abdul Hakim, and Chris Thomas. SteerVLM: Robust model control through lightweight activation steering for vision language models. *arXiv preprint arXiv:2510.26769*, 2025. URL https://arxiv.org/abs/2510.26769.

Samuel Soo, Guang Chen, Wesley Teng, Chandrasekaran Balaganesh, Guoxian Tan, and Ming Yan. Interpretable steering of large language models with feature guided activation additions. *arXiv preprint arXiv:2501.09929*, 2025. URL https://arxiv.org/abs/2501.09929. Building Trust Workshop at ICLR 2025.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders. *arXiv preprint arXiv:2502.06755*, 2025. URL https://arxiv.org/abs/2502.06755. Introduces saev package for training SAEs on ViTs.

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

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C Daniel Freeman, Theodore R Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023. doi: 10.48550/arXiv.2308.10248. URL https://arxiv.org/abs/2308.10248.

Junjie Wen, Minjie Zhu, Jiaming Liu, Zhiyuan Liu, Yicun Yang, Linfeng Zhang, Shanghang Zhang, Yichen Zhu, and Yi Xu. dVLA: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*, 2025. URL https://arxiv.org/abs/2509.25681.

Sihao Wu, Gaojie Jin, Wei Huang, Jianhong Wang, and Xiaowei Huang. Activation steering meets preference optimization: Defense against jailbreaks in vision language models. *arXiv preprint arXiv:2509.00373*, 2025. doi: 10.48550/arXiv.2509.00373. URL https://arxiv.org/abs/2509.00373.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020. URL https://arxiv.org/abs/1910.10897.

Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. Embodied-R1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998*, 2025. URL https://arxiv.org/abs/2508.13998.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting clip with hierarchical sparse autoencoders, 2025. URL https://arxiv.org/abs/2502.20578.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.03954.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. VLM4VLA: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026. URL https://arxiv.org/abs/2601.03309.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2503.22020.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2304.13705.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.10274.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025. URL https://arxiv.org/abs/2510.03827.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Extended Related Work

A.1 Vision-Language-Action Models

RT-2 (Zitkovich et al., 2023) showed that web-pretrained VLMs can be fine-tuned for manipulation via tokenized actions. OpenVLA (Kim et al., 2024) extended this with an open-source 7B model, and OpenVLA-OFT (Kim et al., 2025) replaced discrete tokenization with continuous L1 regression, achieving 97.1% LIBERO success. The π_0 and $\pi_{0.5}$ models (Black et al., 2024; Physical Intelligence, 2025) introduced flow matching with a dedicated action expert. The landscape now spans a range of scales and paradigms: SmolVLA (Shukor et al., 2025) (450M), X-VLA (Zheng et al., 2026) (cross-embodiment), GR00T N1.5 (Bjorck et al., 2025) (humanoid), CogACT (Li et al., 2024a), GR-3 (Cheang et al., 2025), VITA-VLA (Dong et al., 2025), SimpleVLA-RL (Li et al., 2026), RIPT-VLA (Tan et al., 2025), FLOWER (Reuss et al., 2025), SpatialVLA (Qu et al., 2025), and ACT (Zhao et al., 2023) (80M, no language). Modular approaches include SayCan (Ahn et al., 2022), PaLM-E (Driess et al., 2023), and Code as Policies (Liang et al., 2023).

Action Representation. Diffusion Policy (Chi et al., 2023) treats action generation as conditional denoising. 3D Diffusion Policy (Ze et al., 2024) incorporates 3D representations. FAST (Pertsch et al., 2025) uses DCT-based compression tokenization. We find that action tokenization directly determines SAE applicability (Section C.1).

Language Following. CAST (Glossop et al., 2025) augments robot datasets with counterfactual language labels. VLM4VLA (Zhang et al., 2026) found that VLM benchmark performance does not predict VLA task success; language understanding and visuomotor competence are decoupled.

A.2 Mechanistic Interpretability

SAEs (Olshausen & Field, 1997; Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024; Templeton et al., 2024) decompose dense activations into sparse interpretable features, with extensions to vision (Stevens et al., 2025; Joseph et al., 2025; Zaigrajew et al., 2025) and vision-language models (Pach et al., 2025). Activation steering (Turner et al., 2023; Rimsky et al., 2024) enables behavioral control, with advances including conceptor-based steering (Postmus & Abreu, 2024), SAE-guided additions (Soo et al., 2025), and VLM safety steering (Wu et al., 2025; Sivakumar et al., 2025). Linear probing (Alain & Bengio, 2017; Belinkov, 2022) identifies accessible representations but does not establish causal relevance (Mueller, 2024). PvP (Golovanevsky et al., 2025) shows vision and priors compete in multimodal models.

A.3 Interpretability for Robot Learning

RT-1 (Brohan et al., 2022) included attention visualizations, Diffusion Policy (Chi et al., 2023) analyzed action distributions, and ALOHA (Zhao et al., 2023; ALOHA 2 Team et al., 2024) demonstrated bimanual manipulation, all without internal representation analysis. RoboFail (Sagar et al., 2024), AHA (Duan et al., 2024), and RACER (Dai et al., 2025) characterize failures behaviorally. Häon *et al.* (Häon et al., 2025) introduced VLA steering via internal representations. Molinari *et al.* (Molinari et al., 2025) probed for emergent world models. Khan *et al.* (Khan et al., 2025) used SAEs to isolate interpretable steering directions in Magma. Interpretability-by-design approaches include CoA-VLA (Li et al., 2025), CoT-VLA (Zhao et al., 2025), TriVLA (Liu et al., 2025), dVLA (Wen et al., 2025), and Embodied-R1 (Yuan et al., 2025).

B Limitations

Benchmark Scope

While we evaluate on four benchmarks (LIBERO, MetaWorld, SimplerEnv, ALOHA), all experiments use simulated environments. Real-world VLA deployment requires fine-tuning on the target robot and environment, and fine-tuning reshapes internal representations. Whether the architectural-level patterns identified here (visual pathway dominance, spatially bound motor programs, pathway specialization) persist through domain-specific fine-tuning remains untested. Expanding to multi-simulator benchmarks like RoboCasa (Nasiriany et al., 2024), procedurally generated scenes like SceneSmith (Pfaff et al., 2026), real-world datasets like DROID (Khazatsky et al., 2024), and cross-embodiment collections like Open X-Embodiment (Open X-Embodiment Collaboration, 2024) would establish broader generalizability. Robustness evaluations like LIBERO-PRO (Zhou et al., 2025), which shows that models achieving 90%+ standard accuracy collapse to 0% under position perturbations, further motivate interpretability-driven failure analysis.

Counterfactual Prompt Coverage

Our counterfactual prompt set tests simple variations (negation, null, swap). Compositional instructions ("pick up the red cup, then place it behind the blue bowl") and ambiguous multi-object scenes would provide stronger evidence for language insensitivity claims. Benchmarks like LIBERO-PRO (Zhou et al., 2025), which tests corrupted instructions and environmental perturbations, offer a more rigorous evaluation protocol for language grounding.

Cross-Task Injection Confounds

Injected activations can produce invalid internal states due to temporal misalignment rather than reflecting absent abstract representations. However, the displacement analysis strengthens the interpretation: in 99.8% of X-VLA injection episodes, the robot's trajectory is more similar to the source task than the destination, which supports successful source behavior transfer rather than mere distribution shift.

Steering Sensitivity and Concept Identification

Steering sensitivity spans a wide range across architectures: $\pi_{0.5}$ expert pathways are catastrophically sensitive (ΔSR $= -84$pp at $-3\times$ suppression), GR00T DiT features show similar fragility (-68pp at $9\times$ amplification), while OFT (ΔSR $= -6$pp at $-3\times$) and SmolVLA (-3pp at $5\times$ amplification) are comparatively robust. Feature sensitivity is shaped by architecture and training regime rather than being a universal property of VLAs. Three factors contribute: motor control demands sub-millimeter end-effector precision (Chi et al., 2023); sequential action prediction compounds reconstruction errors across 50 tokens (analogous to autoregressive error accumulation (Bengio et al., 2015)); and behavior-cloned policies lack the closed-loop error correction available to RL-trained policies (Ross et al., 2011). Phase-specific steering on $\pi_{0.5}$ shows temporally localized sensitivity during the transport phase ($p = 0.013$, Wilcoxon rank-sum). SAE features are also not guaranteed to be disentangled: ablation validation shows that some identified features encode general motor primitives rather than cleanly disentangled semantic concepts.

C Discrete Tokenization and Vision-Only Control

C.1 Discrete Tokenization Prevents SAE Intervention

Before adopting OpenVLA-OFT, we conducted experiments on base OpenVLA (autoregressive discrete 256-bin tokenization). Despite successful SAE training ($R^2 = 0.87$–0.96), hooking SAEs into the forward pass produced 0% task success on all but the final layer. The discrete tokenization maps activations to bins via argmax; even small reconstruction errors shift the selected bin, with errors compounding across the 7-token autoregressive sequence. Replacing discrete tokenization with continuous L1 regression via OpenVLA-OFT enables SAE intervention at 99.2% success. Action representation, not model scale, determines SAE applicability.

C.2 ACT as Non-VLM Control

ACT provides a critical control, lacking any language pathway. Cross-task injection between TransferCube and Insertion produces outputs *identical* to the uninjected baseline (cosine similarity $= 1.0$, bit-identical action arrays), so encoder representations are entirely task-specific. Grid ablation reveals spatially structured representations: masking grid position (2,2) corresponding to the primary manipulation workspace reduces success from 100% to 10%, while Gaussian noise ($\sigma = 0.1$) is universally devastating (100% $\rightarrow$ 0%). Visual pathway dominance and cross-task failure thus hold even for vision-only policies, which rules out VLM-specific artifacts.

D Extended Methodology Details

D.1 SAE Architecture and Training

Architecture. Our sparse autoencoders consist of an encoder-decoder pair with tied weights:

$$\mathbf{z} = \text{TopK}(\mathbf{W}_e\mathbf{h} + \mathbf{b}_e, k = 64) \tag{1}$$

$$\hat{\mathbf{h}} = \mathbf{W}_e^\top \mathbf{z} + \mathbf{b}_d \tag{2}$$

where $\mathbf{h} \in \mathbb{R}^{1024}$ is the input activation, $\mathbf{W}_e \in \mathbb{R}^{d_h \times 1024}$ is the encoder weight matrix, and $d_h \in \{4096, 8192\}$ is the hidden dimension (4x or 8x expansion).

Training Hyperparameters. We train each SAE on 500,000 activation samples (approximately 10,000 forward passes × 50 action tokens) with a batch size of 4096 for 100 epochs. The learning rate is 3×10^{-4} with cosine decay. Sparsity is enforced via TopK selection with $k = 64$ active features per token. Decoder weights are tied to the encoder transpose ($\mathbf{W}_d = \mathbf{W}_e^\top$).

Per-Token Processing. Each of the 50 action tokens is processed independently through the SAE:

$$\mathbf{h}_{\text{flat}} = \text{reshape}(\mathbf{H}, [B \times 50, 1024]) \tag{3}$$

This preserves the heterogeneous structure of the action token sequence, where early tokens encode initial trajectory direction, middle tokens encode main motion execution, and late tokens encode fine adjustments.

D.2 Concept-Based Feature Identification

We identify concept-associated features using frequency-weighted contrastive selection:

$$\text{score}_f = d_f \times \text{freq}_f \tag{4}$$

where d_f is Cohen's d (Cohen, 1988) measuring activation difference between concept-present and concept-absent tasks, and freq_f is the fraction of samples where feature f appears in the active top-64.

This weighting addresses a methodological consideration: with TopK sparsity, features with high mean activation across samples do not necessarily appear in the active top-64 for individual samples, reducing their causal relevance.

D.3 Ablation Protocol

Feature ablation is performed by zeroing selected features in the SAE latent space:

$$\mathbf{z}_{\text{ablated}} = \mathbf{z} \odot \mathbf{m} \tag{5}$$

$$\mathbf{h}_{\text{modified}} = \mathbf{h} + (\mathbf{W}_e^\top \mathbf{z}_{\text{ablated}} - \mathbf{W}_e^\top \mathbf{z}) \tag{6}$$

where $\mathbf{m} \in \{0, 1\}^{d_h}$ is a binary mask with zeros at ablated feature indices. Per-token ablation applies this independently to each of the 50 action tokens.

E Component-Level Analysis

The interventional experiments in Sections 4.2–4.6 establish causal control over VLA behavior: activation injection overrides task selection, layer zeroing destroys performance, concept ablation produces kill-switches, and feature boosting causes dose-dependent failure across all six models. This section traces information flow at a finer grain: individual activation dimensions, FFN neurons, and layer-to-layer representational similarity. We analyze $\pi_{0.5}$ in depth (FFN weight projection and LDA on the 1024-dim expert space), then validate across OFT (32-layer neuron-level concept mapping, 80 concepts × 11,008 neurons per layer), SmolVLA (FFN neuron contrastive identification on both expert and VLM pathways), and GR00T (per-layer-type SAE feature utilization across DiT, Eagle, and VL-SA).

E.1 Goal Encoding Dimensions

Linear discriminant analysis (LDA) on $\pi_{0.5}$ expert activations at layer 17 identifies three components that separate four goal-suite tasks with 71.9% cross-validated accuracy (n=114, 5-fold). The first component captures 45.8% of discriminant variance, the second 35.2%, and the third 19.0%. The top 20 discriminant dimensions (417, 909, 934, 649, 147, 219, 708, 297, 155, 545, ...) define a low-dimensional subspace sufficient for goal identification within a 1024-dimensional activation space.

Goal information emerges at different rates across pathways. PaliGemma goal classification accuracy rises from 56.4% at layer 0 to 76.4% at layer 13, then declines to 68.9% at layer 17. Expert accuracy is flatter: 64.0% at layer 0, peaking at 66.4% (layer 9), settling at 62.6% by layer 17. PaliGemma's steeper emergence profile confirms its role as the goal encoder; the expert pathway maintains moderate goal awareness throughout but does not refine it.

Individual activation dimensions participate in both goal encoding and action generation. Dimension 62 most strongly modulates the x-component of action output (score 0.074), dimension 618 modulates y (0.071), and dimension 14 modulates z (0.066). The overlap between goal-discriminant dimensions (417, 909, 934) and action-modulating dimensions (62, 618, 14) is minimal, confirming that goal identity and motor execution occupy separable subspaces within the same layer.

Causal validation via subspace injection. We test separability by injecting only the goal-discriminant subspace (20 of 1024 dimensions from LDA) from task A into task B's forward pass at layer 17. Across five task pairs on `libero_goal`, full injection (all 1024 dimensions) causes complete task failure (0% success on 4/5 pairs, down from 100% baseline). Goal-subspace injection (20 dimensions, 2% of the activation space) preserves task success: 100% on three pairs, 67% on a fourth where the source and destination share the OPEN motor primitive. Action-subspace injection (15 dimensions encoding x/y/z action components) shows the same pattern. Replacing 2% of the activation space from a different task does not disrupt the destination task's motor execution, confirming that goal identity and motor programs occupy functionally separable subspaces within the same layer.

E.2 FFN Motor Primitive Neurons

Projecting PaliGemma FFN weight vectors onto a token-association vocabulary identifies neurons that encode specific motor primitives. At layer 17, 6,606 of 16,384 neurons (40.3%) associate with motion verbs (pick, place, move, push, pull), 2,907 (17.7%) with object tokens, 2,989 (18.2%) with gripper state, 2,223 (13.6%) with direction, and 1,151 (7.0%) with spatial relations.

The distribution shifts across layers. Motion verb neurons increase from 4,231 at layer 0 (25.8%) to 6,606 at layer 17 (40.3%), a 56% increase. Gripper neurons triple from 998 (6.1%) to 2,989 (18.2%). Object neurons decrease from 3,660 (22.3%) to 2,907 (17.7%), while spatial neurons decline from 1,655 (10.1%) to 1,151 (7.0%). This progression from object/spatial representation in early layers to motor/gripper representation in late layers traces the computational transformation from scene understanding to action generation within a single pathway.

OFT's 32-layer Llama-2 backbone shows a different pattern: granular concept mapping (80 concepts per layer, 11,008 neurons) reveals that spatial relation neurons (`spatial_in`: 5,483→7,245; `spatial_on`: 4,724→6,727) dominate at all depths and grow monotonically from L0 to L31, while object neurons remain sparse (e.g., `obj_can`: 814→1,175). OFT's 4096-dim representation distributes spatial information broadly rather than concentrating it in late layers, consistent with its resilience to concept ablation (92% zero effect). GR00T's SAE feature analysis reveals a layer-type gradient in feature utilization: DiT L0 retains only 3,654 of 12,288 features (70% dead), rising to 9,317 alive at DiT L15; Eagle L0 has 6,730 alive of 16,384; VL-SA L0 has 11,585 alive. Later DiT layers use more features, consistent with increasing computational demands as the flow-matching denoising process progresses.

Phenomenon	$\pi_{0.5}$	**OFT**	**X-VLA**	**SmolVLA**	**GR00T**	**ACT**
Visual pathway dominance	Y (73%)	Y (14%)	Y (all layers)	Y	Y	Y
Cross-task failure	Y (2.6%)	Y (∼50%)	Y (99.8% src)	Y (16% src)	Y (57%)	Y (0%)
Language sensitivity	suite-indep.	suite-dep.§	suite-dep.§	partial*	suite-dep.	N/A
Pathway specialization	Y	N/A	N/A	Y (2×)	Y	N/A
Per-token SAE req.	Y	Y	reversed†	Y	partial‡	N/A
Causal sensitivity	narrow (54%)	wide (92%)	wide (82%)	narrow (28%)	mixed (59%)	N/A

Table 6: Cross-model validation of core findings. Y = confirmed. § `libero_goal` collapses to 10% under wrong prompts; `libero_object` immune. *SmolVLA: MetaWorld difficulty-dependent. †X-VLA: mean-pooled SAEs achieve better rollout fidelity. ‡GR00T: VL-SA layers benefit from mean-pooling. Causal sensitivity spans 28–92% zero-effect rates.

E.3 Layer Independence via CKA

Centered Kernel Alignment (CKA) between all 18 $\pi_{0.5}$ expert layers reveals near-zero representational similarity between any pair of layers. Consecutive layers share a mean CKA of 0.0007 ± 0.0004; the maximum off-diagonal entry across the entire 18×18 matrix is 0.0023. Every layer performs a substantial transformation on its input; no layer acts as a pass-through or residual relay.

OFT's 32-layer Llama-2 backbone shows a complementary pattern: probing R^2 for episode-length prediction rises from 0.845 (L0) to 0.941 (L24) before settling at 0.915 (L31), while task classification accuracy saturates at 97.7–100% by L8. Information is progressively refined across layers rather than computed in a single step. Combined with the FFN analysis above, these findings confirm that each layer in both architectures contributes a distinct computational step in the scene-to-action pipeline.

E.4 Where Spatial Binding Forms During Fine-Tuning

Comparing backbone activations between base OpenVLA (fine-tuned with discrete action tokens) and OFT (fine-tuned with continuous L1 regression) on the same LIBERO tasks localizes the representational changes induced by fine-tuning. We feed identical images through both models and measure per-layer cosine similarity of mean activations (n=50 images per suite, 4 suites).

Across all four suites, representational divergence follows a monotonic gradient: early/mid layers (L1–L15) remain near-identical (cosine similarity >0.999), divergence increases steadily through L16–L28 (0.997→0.980), and the final layer L31 shows a sharp cliff (0.917–0.938). Layer 0 also dips slightly (0.970–0.989) due to differences in input embedding processing.

This gradient is consistent across suites: `libero_10` shows the largest L31 divergence (cosine 0.917), followed by `libero_object` (0.935), `libero_spatial` (0.936), and `libero_goal` (0.938). Harder suites (10 tasks spanning all manipulation types) induce greater late-layer representational change than single-concept suites.

The concentration of divergence in the final third of the backbone (L20–L31) shows that fine-tuning reshapes late-layer representations to encode action-relevant motor programs while preserving early/mid-layer visual and linguistic features regardless of action head architecture. This localizes spatial binding: the coordinate-specific action sequences identified via cross-task injection (Section 4.3) are encoded in the layers that change most during fine-tuning.

A complementary cross-suite comparison confirms this localization. Comparing four OFT models fine-tuned on different suites (6 pairwise comparisons × 32 layers, n=30 images each), L31 diverges modestly (cosine similarity 0.973–0.992) while mid-layers remain near-identical (L16: 0.998–0.999). The cross-suite L31 divergence (0.973–0.992) is far smaller than the base-vs-OFT divergence (0.917–0.937), confirming that action head architecture (discrete vs. continuous) reshapes late-layer representations more than suite-specific fine-tuning data does.

F Additional Experimental Results

F.1 Per-Token vs Mean-Pooled SAE Reconstruction

Table 7 compares per-token and mean-pooled SAE reconstruction on all LIBERO-10 tasks. Mean-pooled reconstruction achieves comparable explained variance (95-98%) but causes complete task failure: positional information across action tokens is essential for action generation.

Task	Baseline	Per-Token SAE	Mean-Pooled SAE
0	198/250	103/250	2/250
1	247/250	248/250	0/250
2	203/250	198/250	1/250
3	151/250	248/250	0/250
4	249/250	202/250	3/250
5	249/250	247/250	0/250
6	248/250	199/250	2/250
7	153/250	148/250	0/250
8	12/250	8/250	0/250
9	152/250	147/250	1/250
Total	1862/2500	1748/2500	9/2500
Rate	74.5%	69.9%	0.4%

Table 7: Per-task results for per-token vs. mean-pooled SAE reconstruction on LIBERO-10 (2,500 episodes). Per-token SAE maintains task performance (69.9%) while mean-pooled causes near-complete failure (0.4%).

F.2 Layer-Wise Concept Specificity

Table 8 reports the layers with highest feature activation scores for each concept type. Across action, object, and spatial categories, later layers (14–17) show the strongest concept specificity: semantically structured representations emerge progressively through the transformer stack.

Concept	Best Layer	2nd Best	3rd Best
Action Concepts			
PUT	L17 (133k)	L16 (111k)	L15 (86k)
OPEN	L15 (103k)	L16 (79k)	L13 (71k)
PUSH	L17 (412k)	L13 (275k)	L14 (270k)
INTERACT	L17 (274k)	L12 (261k)	L15 (253k)
Object Concepts			
BOWL	L16 (114k)	L15 (80k)	L14 (70k)
WINE_BOTTLE	L16 (128k)	L14 (116k)	L17 (104k)
STOVE	L17 (182k)	L15 (164k)	L12 (141k)

Table 8: Layer-wise concept specificity (activation scores in parentheses). Later layers (14-17) show highest concept specificity.

F.3 Cross-Suite Generalization

Features identified from the Goal suite affect corresponding tasks across Object and Spatial suites (Table 9); concept representations generalize across task variations rather than being narrowly tuned to specific scene configurations. Note that these results use full-layer intervention hooks and absolute effect magnitudes should be interpreted with the caveat described in Section F.3.

Target Suite	Baseline	After PUT Ablation
Goal	100%	5%
Object	100%	0%
Spatial	100%	11%

Table 9: Cross-suite generalization of PUT feature ablation. *Caveat:* These results use full-layer intervention hooks; corrected MLP-targeted ablation shows no significant effect (see Section F.3).

Causal Feature Identification

Linear probes trained to predict action dimensions achieve 97–98% R^2. Projecting out probe directions completely eliminates action prediction (R^2 drops to ≈ 0), so these directions are causally necessary for downstream computation. SAE-based feature interventions reveal a causal asymmetry: ablation of 2–5 concept-associated features produces no statistically significant effect ($p = 0.975$, mean $\Delta = +3.3\%$); the model compensates through redundant representations. Feature boosting, however, produces significant effects: at 7× natural magnitude, success drops by 14% ($p = 2.27 \times 10^{-4}$), and at 15× by 50.7% (Table 13). This asymmetry (tolerance of feature removal but vulnerability to feature addition) fits redundant motor program encoding.

Caveat on Tables 9, 14, 16, and 18. The ablation results in Sections F.3–G.7 were collected using intervention hooks targeting the full layer residual stream rather than the MLP sublayer alone. This stronger intervention disrupts the residual stream's skip connections, producing inflated effect sizes. When hooks are corrected to target only the MLP sublayer, ablation of 2–5 concept-associated features produces no statistically significant effect ($p = 0.975$, mean $\Delta = +3.3\%$). We retain the original tables because the relative patterns they reveal (e.g., temporal criticality of early steps, cross-suite feature transfer, concept specificity) remain informative, but readers should interpret absolute effect magnitudes with caution. Feature boosting results (Table 13) use the corrected MLP-targeted hooks and show significant effects at 7×+ magnitude.

Vision Robustness

Systematic image perturbation across 6,000+ episodes reveals task-dependent visual robustness. We apply horizontal and vertical flips, rotations (90°, 180°, 270°), center crops (50%, 75%), and object-centric crops. Horizontal and vertical flips universally break all models tested (0% success), so spatial orientation is rigidly encoded. Rotation and crop robustness varies by task complexity: simple pick-and-place tolerates mild perturbations while multi-step tasks fail. Object-centric cropping (centering on the manipulation target) outperforms static crops (60% versus 0–20%), which points to reliance on manipulation-relevant regions rather than full scene context.

Table 7 provides per-task breakdowns supporting the per-token requirement described in Section 3.4: mean-pooled SAE reconstruction causes near-complete task failure (0.4% success) despite 95–98% explained variance, while per-token processing maintains 70% success. Table 13 and Figure 11 further quantify steering sensitivity: both dampening ($\alpha < 0$) and boosting ($\alpha > 0$) features cause task failure.

F.4 Concept Feature Discrimination

Table 10 shows that identified concept features achieve high task discrimination: for each concept, the selected features are active precisely on the relevant tasks and inactive on others, achieving 100% binary classification accuracy despite relying on single SAE features.

Concept	Tasks Active	Tasks Inactive	Accuracy
PUT	1,3,4,5,6,7,8	0,2,9	100%
OPEN	0,1	2-9	100%
INTERACT	9	0-8	100%

Table 10: Concept features show high task discrimination.

G Ablation Studies and Negative Results

G.1 SAE Reconstruction Methods

Training SAEs on mean-pooled activations and broadcasting the residual back to all positions causes catastrophic failure (Table 11). Broadcasting a uniform residual corrupts the heterogeneous per-token information required for action generation.

Layer	**Baseline**	**SAE Reconstruction**
Layer 0-11	80-90%	0-5%
Layer 12-17	80-90%	0-5%
action_out_proj_input	93%	76%

Table 11: Mean-pooled SAE reconstruction causes catastrophic failure on all intermediate layers.

G.2 Feature Selection Methods

Selecting features based on correlation with output action dimensions (x, y, z, roll, pitch, yaw, gripper) yields features that are too general, activating across all tasks regardless of concept (Table 12). Action correlations capture output statistics rather than input semantics.

Feature Selection	**Baseline**	**With Ablation**
Action-correlated	93%	79%
Concept-aligned	93%	93% (no ablation)

Table 12: Action-correlated features lack concept specificity.

G.3 Steering Interventions

Negative Results: Recovery, Boosting, and Substitution

Three interventions produce uniformly negative results (Table 13), all using full-layer hooks (see caveat in Section F.3). (1) Steering cannot recover from ablation: boosting the same features at α=0.5–1.0 after ablation produces 0% success across all conditions. Once the first forward pass is corrupted, errors compound. (2) Feature boosting is bidirectionally destructive: both dampening (α=-0.5, 5.7% concept / 0% other) and boosting (α=+1.0, 5.7% / 13.3%) cause near-total failure from a 97.1% baseline. (3) Concept substitution fails: ablating OPEN features and boosting PUT features produces 0% success, confirming that each concept occupies a distinct subspace.

Intervention	**Condition**	**SR**
Recovery	Ablate full episode	0%
	Ablate + steer α=0.5	0%
	Ablate first 50 + steer	0%
Boosting	Dampen α=-0.5	5.7%
	Boost α=+1.0	5.7%
Substitution	Ablate OPEN + boost PUT 0.5	0%
	Ablate OPEN + boost PUT 1.0	0%

Table 13: Negative results: ablation recovery, feature boosting, and concept substitution all fail. Baseline: 97–100%. Recovery and substitution use full-layer hooks (Section F.3); boosting uses corrected MLP-targeted hooks.

G.4 Temporal Ablation Patterns

Ablation effects vary by episode phase (Table 14). Features are critical during early and mid phases (approach and manipulation) but have minimal effect during late phases (placement/release).

Ablation Window	Avg Effect	Phase
step0	∼0%	Episode start
early (0-50)	**-56%**	Approach/grasp
mid (50-150)	**-57%**	Manipulation
late (150-300)	-1%	Placement
full (0-300)	**-76%**	All phases

Table 14: $\pi_{0.5}$ temporal ablation effects averaged across all concepts tested. *Caveat:* These results use full-layer intervention hooks; absolute effect magnitudes are inflated (see Section F.3).

Table 15 shows GR00T N1.5 temporal ablation across three LIBERO suites (160 conditions, MLP-targeted hooks). On DiT layers, early-window ablation (-50.3pp, 58% destructive) is nearly as severe as full-episode ablation (-50.8pp), while mid- and late-window ablation causes only -12pp. This temporal asymmetry confirms that motor programs are committed during trajectory initiation: once past the approach phase, DiT features become expendable. Eagle LM layers show a flatter profile (early -15.1pp vs. late -11.8pp), consistent with their role in sustained task context. The effect scales with task complexity: `libero_long` (303-step mean) shows -62pp early-DiT drop vs. -44pp on `libero_goal` (108-step mean).

Suite	Baseline	Full	Early	Mid	Late
LIBERO-Goal	100%	73.1%	73.2%	92.7%	92.4%
LIBERO-Long	100%	53.8%	55.6%	75.3%	75.9%
LIBERO-Object	100%	93.9%	94.0%	95.0%	95.3%
Average	**100%**	**69.5%**	**70.3%**	**86.2%**	**86.4%**

Table 15: GR00T N1.5 temporal ablation by LIBERO suite (160 conditions across 32 layers). Averages are weighted by number of conditions per suite. Early-phase ablation causes the largest success drop, confirming temporal criticality across architectures.

G.5 Single Feature Ablation

Ablating a single feature (3259) causes complete task failure (Table 16), and adding more ablated features produces no further degradation, consistent with the single-point-of-failure behavior described for $\pi_{0.5}$ in Section 4.6. As noted in Section F.3, these results use full-layer intervention hooks.

Features Ablated	Success Rate
0 (baseline)	90%
1 (feature 3259)	0%
2	0%
5	0%

Table 16: Single feature ablation causes complete task failure. *Caveat:* These results use full-layer intervention hooks; corrected MLP-targeted ablation shows no significant effect (see Section F.3).

G.6 Step 0 Criticality

Ablating only step 0 causes complete failure, while later steps show minimal impact (Table 17). The first forward pass commits the robot to a trajectory.

Ablation Timestep	Success Rate
Step 0 only	0%
Step 1 only	100%
Step 2 only	80%
Steps 200+	80%

Table 17: Step 0 is uniquely critical for task success.

G.7 Feature Specificity Analysis

Some concept-associated features encode motor primitives shared across tasks rather than task-specific semantics (Table 18).

Concept	Selectivity	Type
PUSH	+8.9%	Task-specific
WINE_BOTTLE	+7.5%	Object-specific
PUT	-6.7%	Motor primitive
OPEN	-10.0%	Motor primitive
INTERACT	-11.1%	Motor primitive

Table 18: Selectivity = (effect on concept tasks) − (effect on other tasks). Negative selectivity indicates features encode motor primitives affecting all tasks. *Caveat:* These results use full-layer intervention hooks; absolute magnitudes are inflated (see Section F.3).

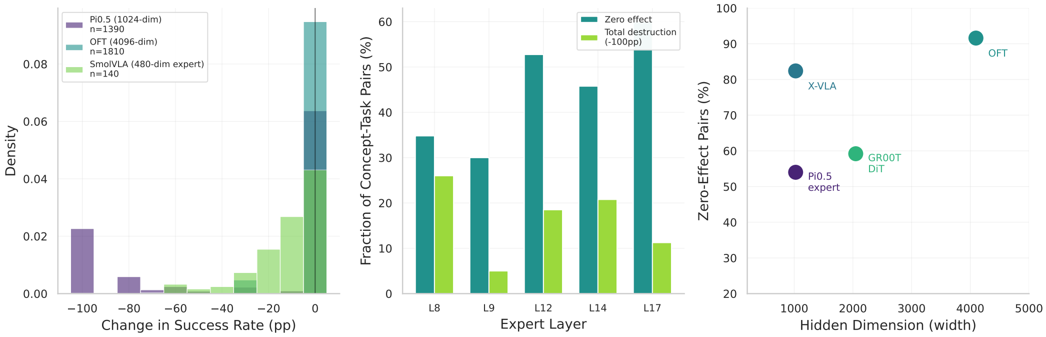


Figure 4: **Concept ablation causal sensitivity across five models.** Each bar shows the fraction of concept-task pairs with zero effect (gray), partial effect (blue), and total destruction (−100pp, red) under single-feature ablation. SmolVLA (480-dim expert) is the most sensitive at 28% zero-effect rate; OFT (4096-dim) and X-VLA (1024-dim) are the most resilient at 92% and 82% respectively. Causal sensitivity does not follow representation width: X-VLA approaches OFT despite sharing $\pi_{0.5}$'s 1024-dim hidden size.

H Benchmark Details

LIBERO (Liu et al., 2023) comprises four suites of 10 tasks each: **Goal** (long-horizon goal completion), **Object** (pick-and-place with varied objects), **Spatial** (spatial reasoning and relational placement), and **LIBERO-10** (diverse tasks spanning all three categories). MetaWorld (Yu et al., 2020) provides 50 tabletop manipulation tasks grouped by difficulty (easy, medium, hard, very hard). SimplerEnv (Li et al., 2024b) evaluates sim-to-real transfer on Google Robot and WidowX embodiments. ALOHA (Zhao et al., 2023) tests bimanual manipulation with the ACT policy.

I Qualitative Results and Additional Figures

Figures 5–8 show frame sequences from baseline and ablated rollouts. All concept ablation uses full-layer hooks (see caveat in Section F.3). Across all conditions, feature ablation and vision perturbation produce binary failure: the robot either completes the task or fails entirely, with no partial completion observed.

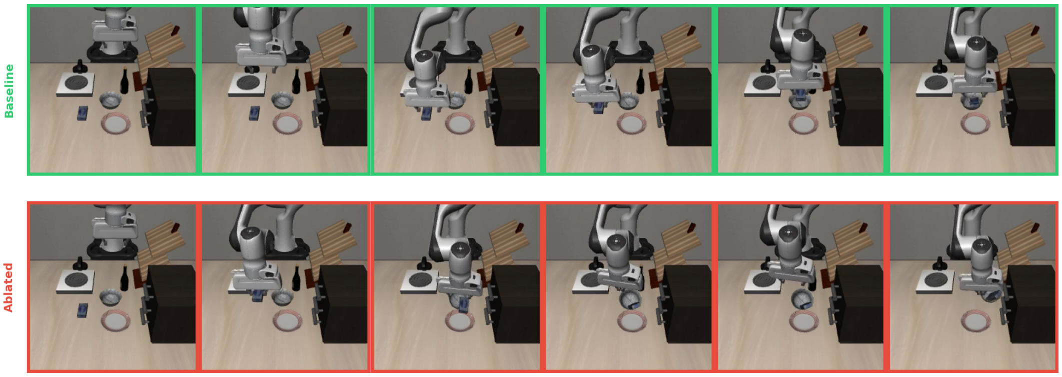


Figure 5: **PUT concept ablation (L8): "Put the cream cheese in the bowl." Top (green):** Baseline. The robot picks up the cream cheese and places it in the bowl (91 steps). **Bottom (red):** With PUT features zeroed at layer 8, the robot drops the cream cheese into the bowl, knocking it over (300 steps, task failure).

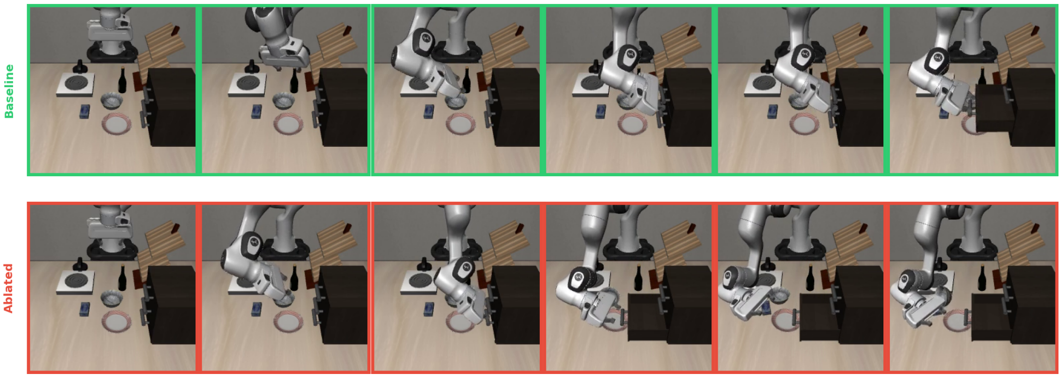


Figure 6: **OPEN concept ablation (L8): "Open the middle drawer of the cabinet." Top (green):** Baseline. The robot reaches for the middle drawer handle, grasps it, and pulls it open (140 steps). **Bottom (red):** With OPEN features zeroed at layer 8 (40% destruction rate), the robot opens the bottom drawer instead of the middle, misdirecting the motor program to the wrong target (300 steps, task failure).

J Per-Suite Experimental Breakdowns

OpenVLA-OFT Per-Suite Results

Table 19 shows null injection results on OpenVLA-OFT. Zeroing any single layer (0, 8, 16, 24, or 31) causes catastrophic failure across all four LIBERO suites, with aggregate success rates of 14–15%.

OpenVLA-OFT Multi-Layer Probing

Linear probes trained on OFT's 32-layer activations (n=149 episodes per suite) reveal that episode-length information (R^2) is distributed across all layers but varies by suite complexity.

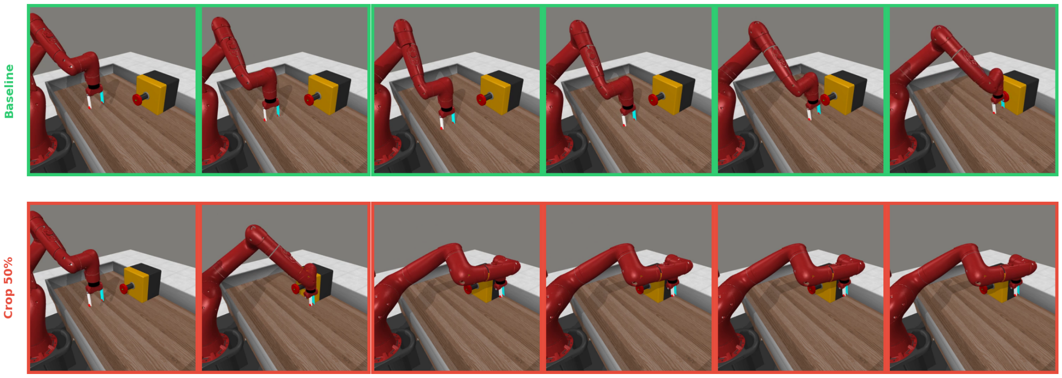


Figure 7: **SmolVLA MetaWorld vision perturbation (button-press).** Top: baseline (67 steps, success). Bottom: center crop to 50% of the visual field removes spatial context; the robot arm cannot locate the button and times out at 400 steps.

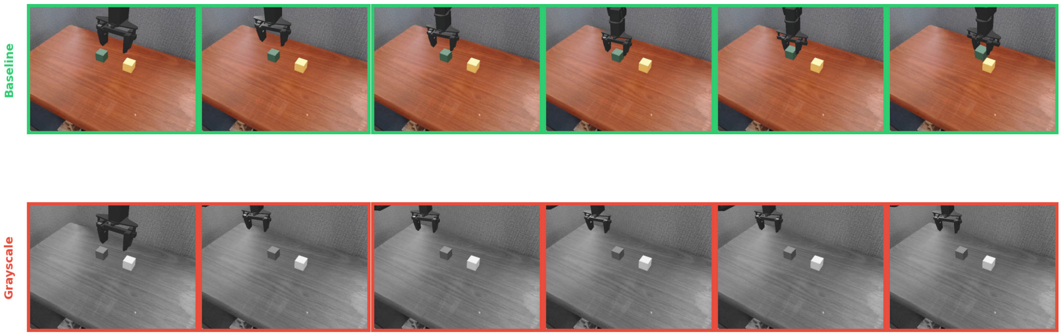


Figure 8: **X-VLA SimplerEnv vision perturbation (stack cube). Top (green):** Baseline WidowX stacks the cube (100% success). **Bottom (red):** Grayscale removes color information, causing complete failure (0% success). The robot cannot distinguish the cube from the table surface without color.

LIBERO-Spatial achieves uniformly high R^2 (0.969–0.994, mean 0.988), while LIBERO-Object and LIBERO-10 show lower floors (0.608 and 0.692) with a progressive increase from early to late layers. Task identification accuracy reaches 97.8–100% on every layer of every suite, and success prediction AUC is 1.0 on spatial, goal, and object suites (mean 0.97 on LIBERO-10, where some early layers drop to 0.71). This confirms that OFT's 4096-dim representations encode sufficient information for linear action prediction at all layers, consistent with its wide, resilient causal profile (92% zero-effect rate).

Cross-Model Linear Probing Summary

Table 24 summarizes linear probe and oracle probe results across all models where probing experiments were conducted. Probes are trained on layer activations (or SAE features for GR00T) to predict task identity, success, state information, or prompt category. The results confirm that task-relevant information is linearly decodable across all architectures, but the type of information encoded differs by pathway: expert/action pathways encode state dynamics while VLM pathways encode goal semantics.

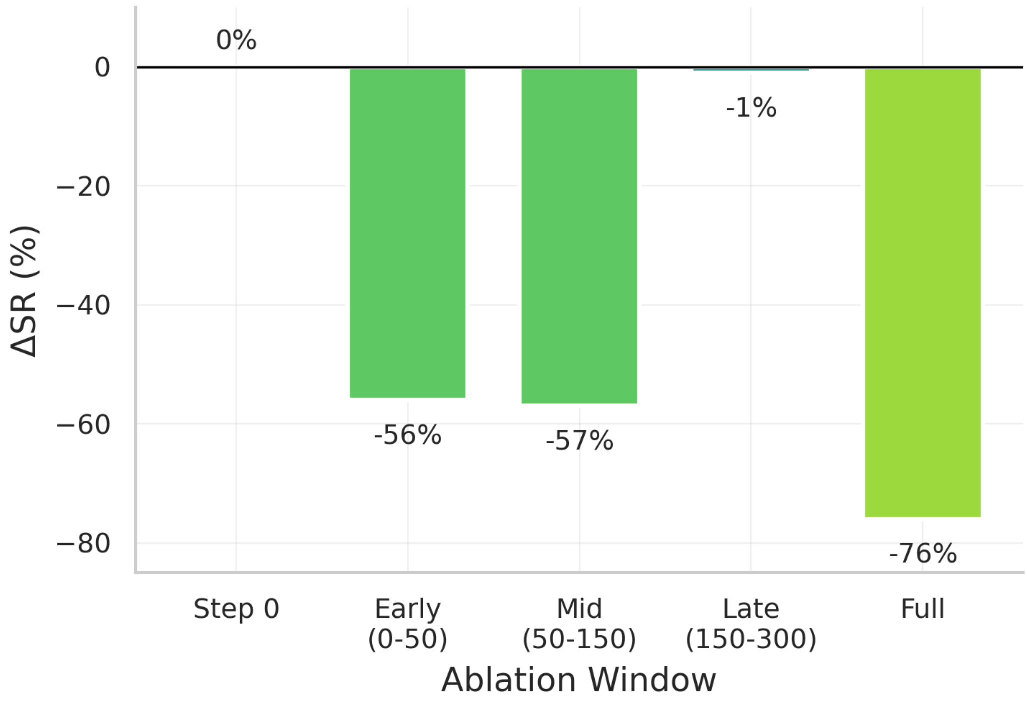


Figure 9: $\pi_{0.5}$ **temporal ablation effects by episode phase.** Feature ablation effects vary by phase using full-layer intervention hooks (see note in Section F.3). Corrected MLP-targeted ablation shows no significant temporal pattern ($p = 0.975$); the phase-dependence shown here is an artifact of the uncorrected hook rather than a genuine temporal gradient. For valid cross-architecture temporal results, see GR00T temporal ablation in Table 15.

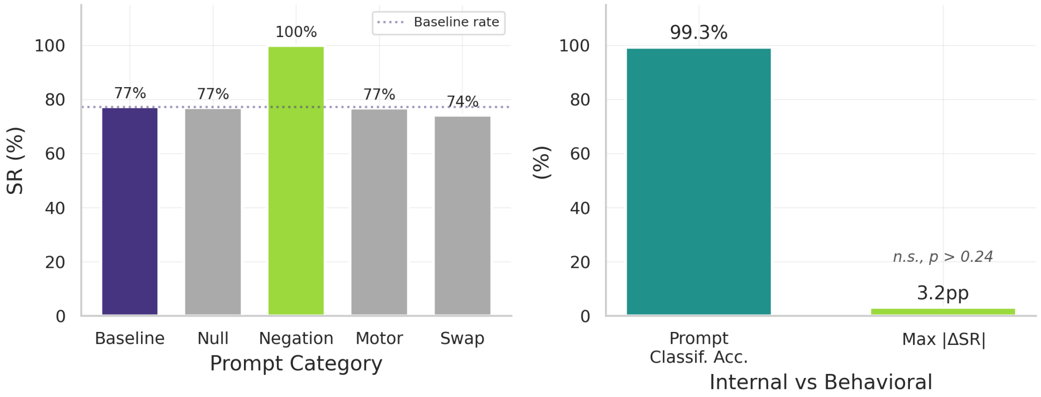


Figure 10: **Language is ignored despite internal distinction.** Left: counterfactual prompting across 3,396+ episodes shows no significant behavioral difference (p>0.24). Right: layer 17 classifiers distinguish prompts with 99.3% accuracy, yet behavior is unchanged.

PI0.5 PER-SUITE COUNTERFACTUAL PROMPTING

PI0.5 CROSS-TASK INJECTION PER-SUITE

X-VLA PER-SUITE RESULTS

X-VLA baselines average 96.7% (goal), 100% (object), 90.0% (spatial), and 100% (LIBERO-10) across 10 tasks per suite (n=3 episodes per task). Table 27 summarizes grid ablation, counterfactual prompting, and concept ablation by suite.

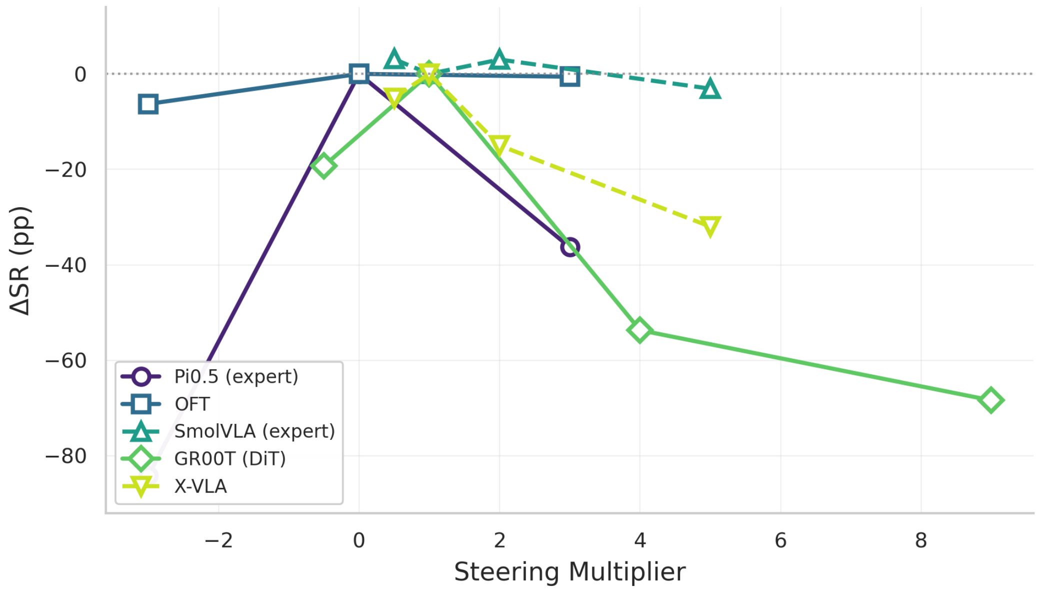


Figure 11: **Steering sensitivity across five models.** $\pi_{0.5}$ and GR00T exhibit catastrophic sensitivity to feature steering, while OFT and SmolVLA are comparatively robust. SmolVLA shows non-monotonic dose response (mild amplification at 0.5–2×, degradation only at 5×). Error represents mean ΔSR across all steered features.

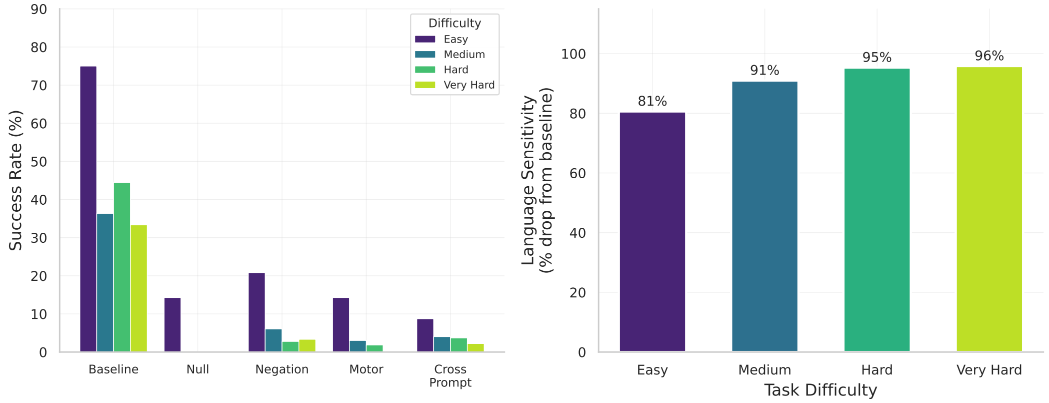


Figure 12: **SmolVLA language sensitivity varies by MetaWorld task difficulty.** Baseline success rates range from 85% (easy) to 62% (hard). Under null prompts, easy tasks drop only 3pp (85%→82%), while hard tasks drop 21pp (62%→41%), showing that language sensitivity scales with task ambiguity rather than architecture.

X-VLA exhibits the strongest suite-dependent language sensitivity: `libero_goal` collapses from 94% to 10% under null prompts, while `libero_object` retains 60%. Concept ablation is uniformly resilient across suites (74.7–98.2% zero effect), with `libero_object` nearly immune (98.2% zero effect, −1.3pp mean delta).

SmolVLA Per-Suite Results

SmolVLA LIBERO baselines: goal 75.0%, object 78.5%, spatial 67.5%, LIBERO-10 40.7% (n=20 per task). Grid ablation across 65 conditions (32 expert + 32 VLM + baseline) shows that expert-layer zeroing maintains partial success (Table 28), while concept ablation reveals stronger sensitivity than OFT or X-VLA.

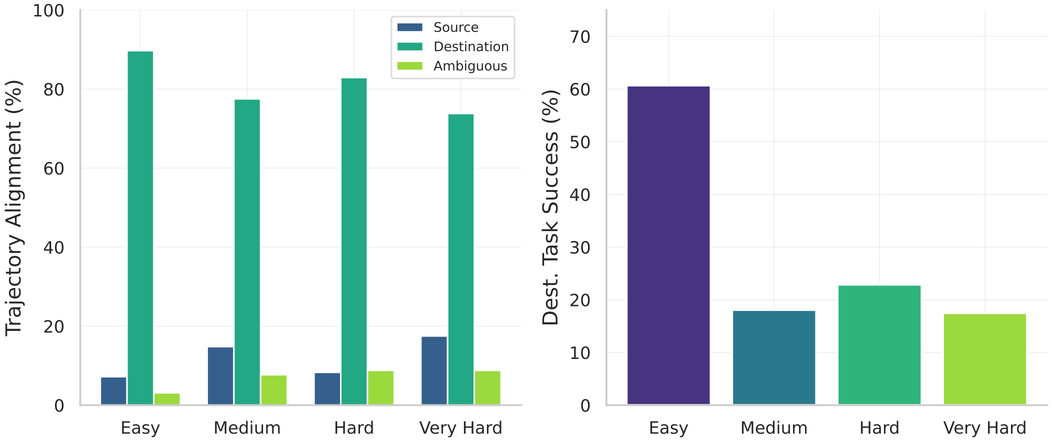


Figure 13: **SmolVLA MetaWorld cross-task displacement and success under injection.** Left: source-task override rate by MetaWorld difficulty level: destination alignment decreases as difficulty increases (89.7% on easy, 73.8% on very hard), so harder tasks produce weaker source behavior transfer. Right: destination task success under cross-task injection by difficulty: easy tasks maintain 60.7% success while harder tasks drop to 17–22%, reflecting both the displacement effect and the greater task-specific precision required for harder goals.

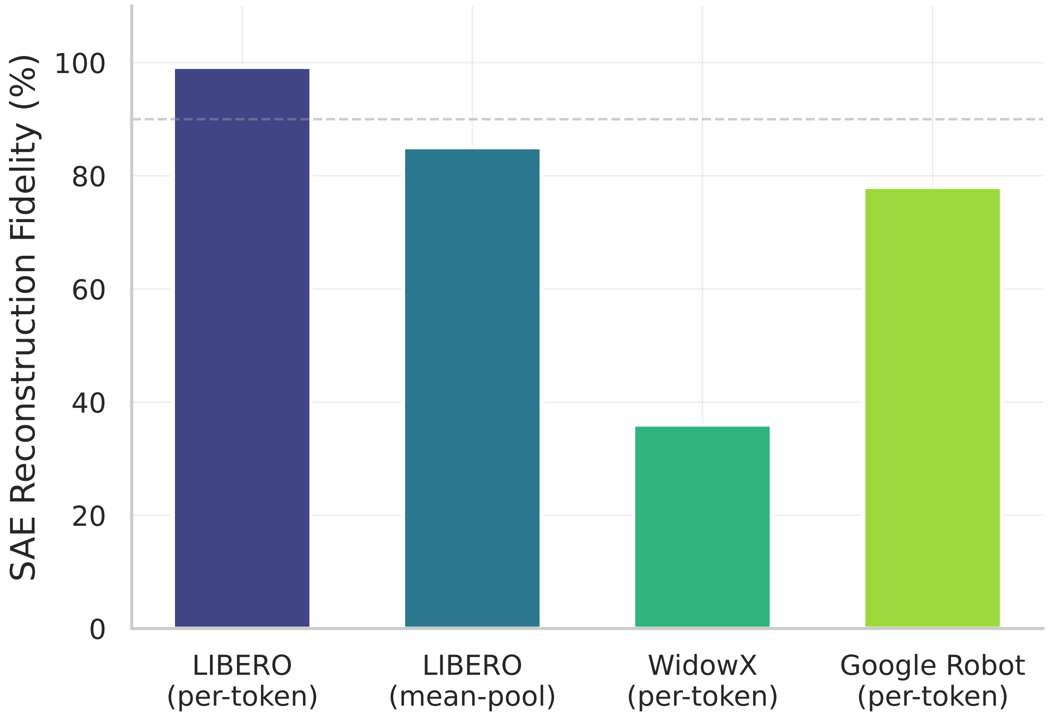


Figure 14: **X-VLA SAE reconstruction fidelity across environments.** Per-token processing achieves 99.2% explained variance on LIBERO but only 36% on WidowX (SimplerEnv), while mean-pooling reduces LIBERO fidelity to 85%. The cross-environment degradation indicates that SAE features trained on LIBERO activations do not fully transfer to the WidowX embodiment distribution.

SmolVLA expert concept ablation is most destructive on `libero_spatial` (−17.7pp, 32.7% zero effect) and least on `libero_10` (+0.8pp, 20.3% zero effect). VLM concept ablation shows

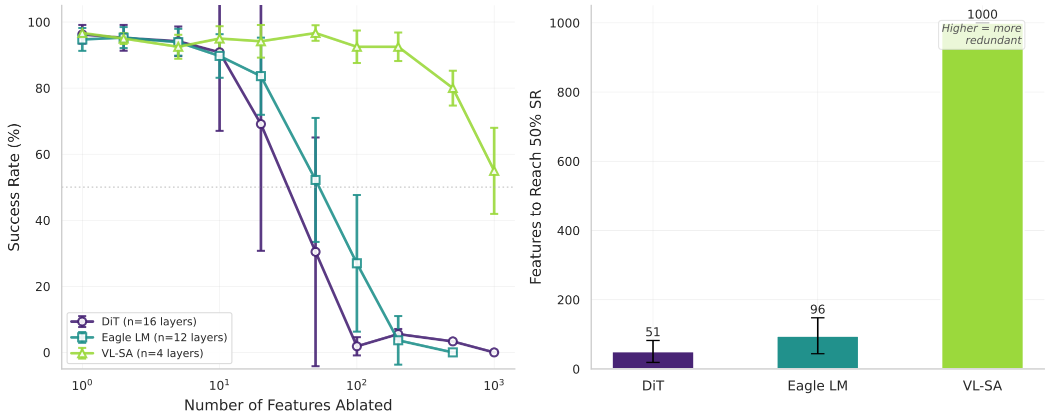


Figure 15: **GR00T N1.5 layer-type contribution profiles.** SAE feature ablation across 32 layers (16 DiT + 12 Eagle LM + 4 VL-SA) reveals distinct importance profiles per layer type: DiT layers are the most ablation-sensitive (40–80% success drop), Eagle LM layers show moderate sensitivity, and VL-SA layers are the most resilient. This mirrors the pathway specialization observed in $\pi_{0.5}$ and SmolVLA.

Suite	Baseline	Null Layer	Aggregate
libero_goal	~90%	~0%	33/234 (14%)
libero_object	~90%	~0%	43/294 (15%)
libero_spatial	~90%	~0%	31/216 (14%)
libero_10	9/10 tasks	0/10 tasks	–

Table 19: OpenVLA-OFT null injection by suite. Zeroing any single layer (tested: 0, 8, 16, 24, 31) destroys task success across all suites.

a different pattern: `libero_goal` is most affected (−10.7pp) while `libero_object` shows positive delta (+6.4pp), consistent with the VLM encoding goal semantics.

GR00T N1.5 Per-Suite Results

GR00T grid ablation baselines: goal 97.0%, object 99.0%, long 75.0% (n=10 per task; differs from the null injection baselines in Table 3 due to separate experiment batches with different n). Table 29 summarizes grid ablation, counterfactual prompting, and concept ablation.

GR00T mirrors X-VLA's suite-dependent language sensitivity: `libero_goal` collapses from 96.7% to 18.9% under non-baseline prompts, while `libero_object` retains 73.3%. LIBERO-Long shows intermediate sensitivity (61.7%), consistent with its multi-step structure requiring partial language grounding. Concept ablation reveals that `libero_long` has the lowest zero-effect rate (42.2%) and highest destruction rate (11.0%), reflecting the greater fragility of complex multi-step tasks.

K Implementation Details

K.1 Model Architecture

Pi0.5 Architecture

$\pi_{0.5}$ uses PaliGemma (3B parameters) as its vision-language backbone and an 18-layer Gemma transformer (1024 hidden dimension) as the action expert. The action space is 7-dimensional (dx, dy, dz, dax, day, daz, gripper). The model generates 50 actions per forward pass through flow matching with iterative denoising.

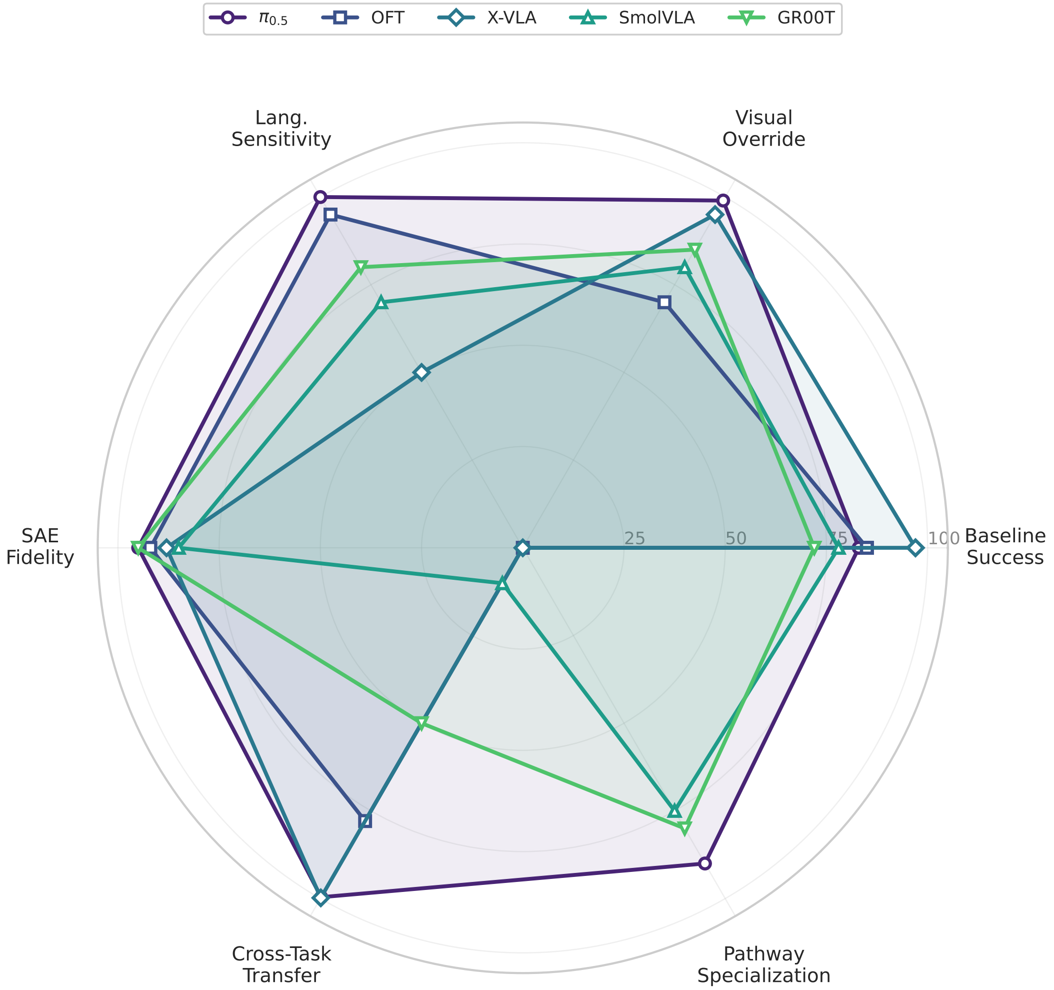


Figure 16: **Cross-model capability radar.** Five VLAs scored on baseline success, visual override strength, language sensitivity, SAE fidelity, cross-task transfer rate, and pathway specialization. OFT lacks pathway specialization (single-pathway architecture); SmolVLA and GR00T show the strongest pathway specialization alongside $\pi_{0.5}$.

Suite	Episodes	Aggregate Success
libero_goal	582	87.8%
libero_object	162	77.8%
libero_spatial	192	74.0%
libero_10	582	71.8%

Table 20: OpenVLA-OFT same-scene injection by suite (1,518 total episodes).

Base OpenVLA Architecture (Discrete)

Base OpenVLA uses a Llama-2 7B backbone with DINOv2 and SigLIP vision encoders. Actions are generated autoregressively as discrete tokens: each of the 7 action dimensions is independently binned into 256 discrete values, producing a 7-token sequence generated left-to-right via next-token prediction.

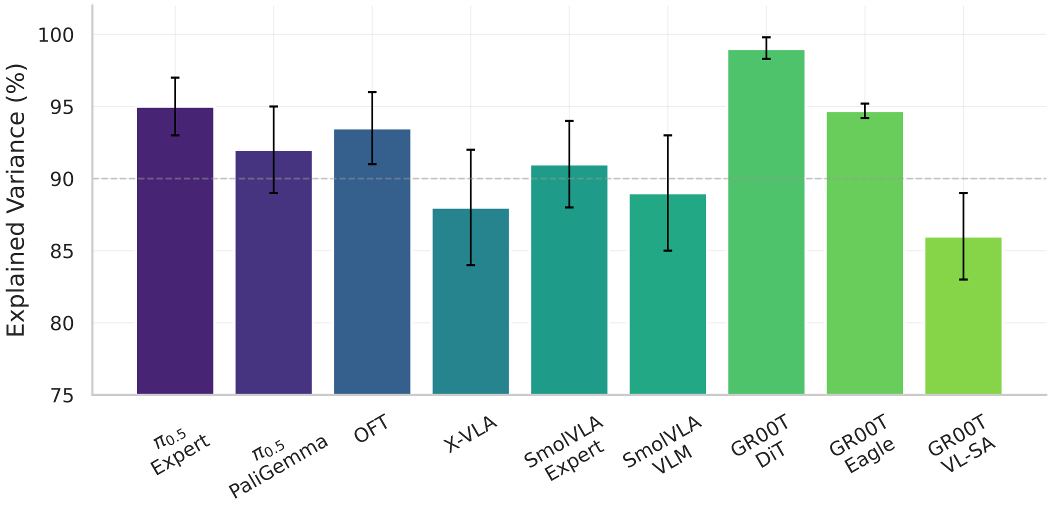


Figure 17: **SAE explained variance by architecture and pooling strategy.** GR00T DiT layers achieve the highest per-token reconstruction quality (98.3–99.8% EV); GR00T VL-SA layers show the lowest per-token quality (83–89%) but improve to 99% EV with mean-pooling. X-VLA mean-pooled SAEs achieve better rollout fidelity than per-token despite lower EV, likely due to Florence-2's more uniform token representations. Error bars: range across layers within each model. Dashed line: 90% threshold.

Window	Task Survival	Surviving Tasks
Baseline	9/10	Tasks 0–5, 7–9
null_early	0/9	None
null_mid	0/9	None
null_late	2/9	Tasks 3, 9

Table 21: OpenVLA-OFT temporal injection on `libero_10`. Early and mid-episode nulling destroys all tasks; late nulling allows partial recovery.

OpenVLA-OFT Architecture (Continuous)

OpenVLA-OFT shares the Llama-2 7B backbone and Prismatic vision encoder (DINOv2 + SigLIP) with the base model but replaces discrete token prediction with continuous action regression. An MLPResNet action head, trained with L1 loss, generates all 7 action dimensions in a single forward pass with 8-step action chunking (56 action tokens total). Optimized Fine-Tuning (OFT) uses LoRA adapters for parameter-efficient adaptation while preserving the pretrained representation geometry.

ACT Architecture

ACT uses a ResNet-18 vision encoder (pretrained on ImageNet) feeding into a Transformer Encoder-Decoder with a CVAE latent space ($\beta = 10$, latent dimension 32). The encoder processes multi-view camera observations, and the decoder generates action chunks of 100 timesteps in 14-DOF joint space (7 per arm for bimanual manipulation).

SAE Configuration

SAE input dimensions match each model's residual stream width: 1024 for $\pi_{0.5}$ expert and X-VLA, 4096 for OFT, 960/480 for SmolVLA VLM/expert, and architecture-dependent for GR00T (DiT, Eagle, VL-SA). The SAE hidden dimension is set to 4× or 8× expansion. Sparsity is enforced via TopK with $k = 64$ active features per token, and decoder weights are tied to the encoder transpose ($\mathbf{W}_d = \mathbf{W}_e^\top$).

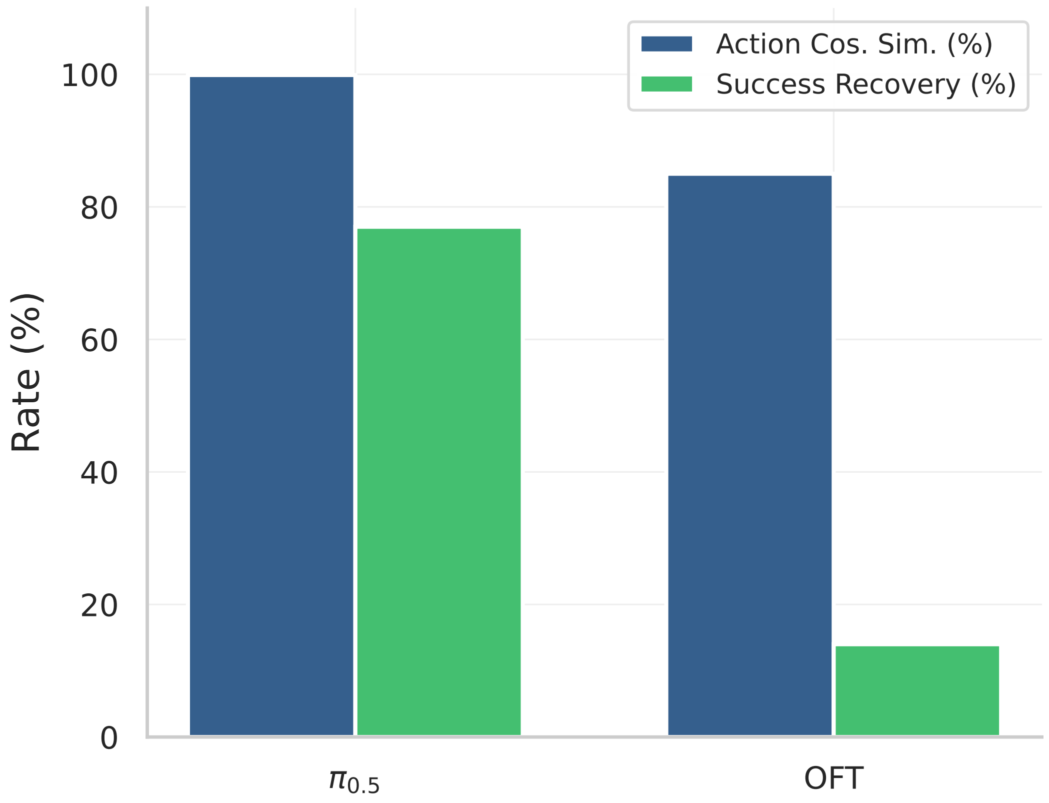


Figure 18: **Null injection recovery for $\pi_{0.5}$ and OFT.** Action cosine similarity (left bars) measures how closely injected actions match baseline; success recovery (right bars) measures task completion under null prompt with injected baseline activations. $\pi_{0.5}$ recovers 77% success with 0.999 cosine similarity, showing strong visual pathway dominance. OFT recovers only 14% despite 0.85 cosine similarity; OFT's representations are more entangled with language context.

Suite	**Min R^2**	**Max R^2**	**Mean R^2**	**Layers $>$ 0.70**
LIBERO-Spatial	0.969	0.994	0.988	32/32 (100%)
LIBERO-Goal	0.845	0.941	0.896	32/32 (100%)
LIBERO-10	0.692	0.837	0.780	30/32 (93.8%)
LIBERO-Object	0.608	0.885	0.813	29/32 (90.6%)
All suites	**0.608**	**0.994**	**0.869**	**123/128 (96.1%)**

Table 22: OpenVLA-OFT multi-layer probing: episode-length R^2 across all 32 layers for four LIBERO suites (n=149 per suite).

K.2 Compute Requirements

Experiments were conducted on an 8×A100-SXM4-80GB cluster for $\pi_{0.5}$, OpenVLA-OFT, large-scale SAE training, concept ablation, and cross-model analysis, an NVIDIA RTX 5090 (32GB) for GR00T N1.5 experiments, and two NVIDIA RTX 4090s (24GB) for SmolVLA and X-VLA experiments. Total experiment data exceeds 7.1 TB, including 4.3 TB of activation recordings, 152 GB of SAE checkpoints, and 394,000+ rollout episodes with videos.

Suite	**Early (L0–7)**	**Middle (L8–23)**	**Late (L24–31)**
LIBERO-Spatial	0.984	0.991	0.986
LIBERO-Object	0.719	0.834	0.863
LIBERO-Goal	0.859	0.896	0.933
LIBERO-10	0.733	0.779	0.827

Table 23: OpenVLA-OFT multi-layer probing: mean R^2 by layer region.

Model	**Probe Target**	**Metric**	**Result**
$\pi_{0.5}$ (expert)	State prediction	R^2	0.45
$\pi_{0.5}$ (expert)	Success prediction	AUC	0.93
$\pi_{0.5}$ (PaliGemma)	Goal classification	Accuracy	76.4%
$\pi_{0.5}$ (PaliGemma)	Prompt classification	Accuracy	99.3%
$\pi_{0.5}$ (PaliGemma)	Workspace region (L15)	Accuracy	99.3%
OFT	Task identification	Accuracy	97.8–100%
OFT	Success prediction	AUC	0.97
OFT	Episode-length (mean)	R^2	0.87
SmolVLA (expert)	Oracle state ratio (h=10)	ratio	0.58
SmolVLA (VLM)	Oracle state ratio (h=10)	ratio	0.13
GR00T (all layers)	Task identification	Accuracy	100%
GR00T (all layers)	Success prediction	Accuracy	96.4%
GR00T (DiT L14)	Success prediction	Accuracy	97.7%

Table 24: Cross-model linear probing summary. $\pi_{0.5}$ expert activations encode state and success while PaliGemma encodes prompt/goal semantics despite behavioral invariance. OFT achieves near-perfect task identification and success prediction. SmolVLA expert captures $4.5\times$ more state information than VLM (0.58 vs. 0.13 oracle ratio). GR00T SAE features achieve perfect task identification across all 32 layers. Oracle ratio: probe R^2 / oracle R^2 (fraction of ground-truth state information linearly decodable from activations).

K.3 Evaluation Protocol

Each experimental condition is evaluated over 5 episodes per task. Episodes run for a maximum of 300 steps. Success is determined by task-specific criteria defined in each benchmark (LIBERO, MetaWorld, SimplerEnv, ALOHA).

L Action Atlas Visualization Platform

We release Action Atlas as an open-source visualization platform for exploring VLA concept representations, available at https://action-atlas.com. The platform provides interactive tools for SAE feature exploration, semantic search, LLM-based auto-interpretability, and ablation video comparisons.

Suite	**Episodes**	**Categories**	**ANOVA** p
libero_object	1,496	22	> 0.24
libero_spatial	353	9	> 0.067
libero_goal	380	6	> 0.24

Table 25: $\pi_{0.5}$ counterfactual prompting by suite. No suite shows statistically significant behavioral differences across prompt categories.

Suite	**Condition**	**Success**	**n**
GOAL	own_prompt (baseline)	91.5%	236
	cross_prompt, no inject	0.0%	236
	cross_prompt + PaliGemma ALL	0.4%	236
	cross_prompt + Expert L16	1.3%	236
SPATIAL	own_prompt (baseline)	100%	34
	cross_prompt, no inject	64.7%	34
	own_prompt + PaliGemma ALL	20.6%	34
LIBERO-10	own_prompt (baseline)	62.5%	24
	all injection conditions	0.0%	24

Table 26: $\pi_{0.5}$ cross-task injection with pathway-specific conditions.

Suite	**Baseline**	**Zero any layer**	**Null prompt**	**Zero eff.**	**Δpp**
Goal	96.7%	0%	10%	82.6%	-2.2
Object	100%	0%	60%	98.2%	-1.3
Spatial	90.0%	0%	48%	85.2%	-2.6
LIBERO-10	100%	0%	28%	74.7%	-2.4

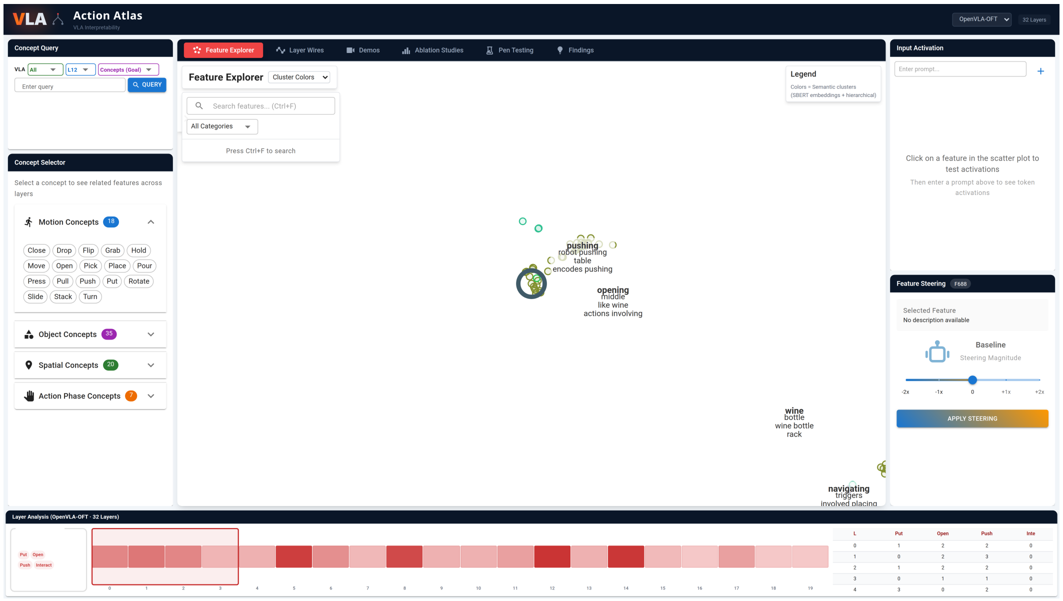
Table 27: X-VLA per-suite breakdown. *Zero any layer*: zeroing any single layer (24 tested) produces 0% success across all suites. *Null prompt*: success under empty string. *Zero eff.*: concept ablation zero-effect rate. Δ*pp*: mean concept ablation delta in percentage points. Counterfactual: n=50 baseline + n=1,100 "other" per suite (n=4,800 total). Concept ablation: n=2,480 pairs total.

Suite	**Baseline**	**Expert zero**	**Zero eff. (exp)**	**Δpp (exp)**
Goal	75.0%	60–83%	14.6%	-4.7
Object	78.5%	60–83%	0.0%	-3.9
Spatial	67.5%	47–77%	32.7%	-17.7
LIBERO-10	40.7%	0–33%	20.3%	$+0.8$

Table 28: SmolVLA per-suite breakdown. *Expert zero*: range of overall success rates when zeroing individual expert layers. Zero eff./Δpp: concept ablation statistics for the expert pathway (n=1,696 pairs total). VLM concept ablation (n=210 pairs): goal -10.7pp, object $+6.4$pp, spatial -8.5pp, LIBERO-10 $+5.9$pp.

Suite	**Baseline**	**Zero DiT**	**CF "other"**	**Zero eff.**	**Δpp**
Goal	97.0%	0%	18.9%	68.8%	-5.8
Object	99.0%	0%	73.3%	69.6%	-9.9
Long	75.0%	0%	61.7%	42.2%	-5.3

Table 29: GR00T N1.5 per-suite breakdown. *Zero DiT*: zeroing any DiT layer (16 tested) produces 0% on all suites. *CF "other"*: counterfactual success under non-baseline prompt categories (n=180 per suite). *Zero eff./Δpp*: concept ablation across 6,500 pairs total. LIBERO-Long is most sensitive (42.2% zero effect, 11.0% destruction) reflecting its multi-step task complexity.



Enter query

Search features... (Ctrl+F)

Press Ctrl+F to search

Colors = Semantic clusters (SBERT embeddings + hierarchical)

Enter prompt...

Click on a feature in the scatter plot to test activations

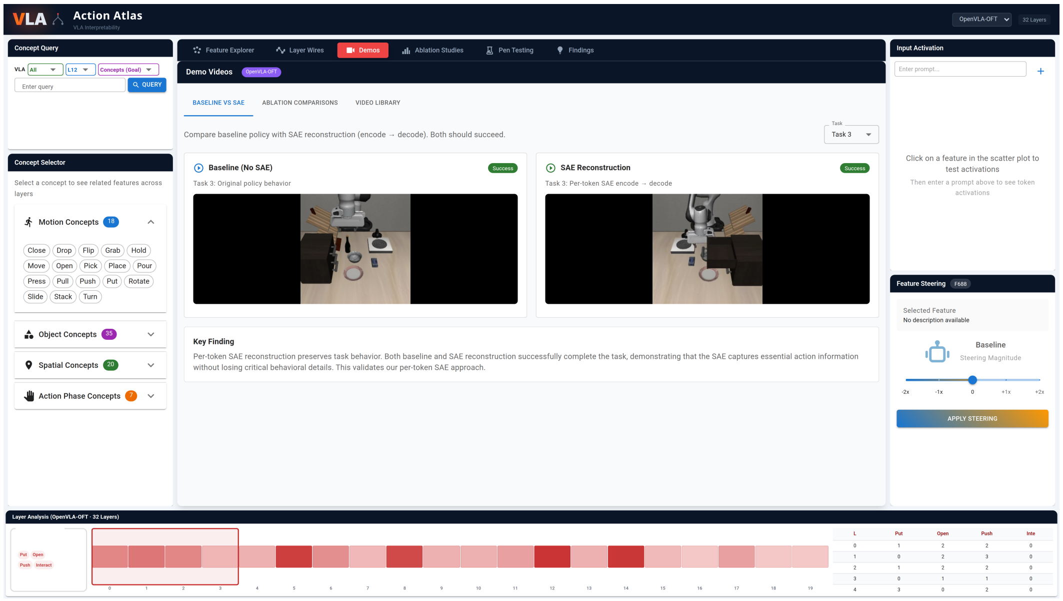
Then enter a prompt above to see token activations

Select a concept to see related features across layers

Selected Feature
No description available

L	Put	Open	Push	Inte
0	1	2	2	0
1	0	2	3	0
2	1	2	2	0
3	0	1	1	0
4	3	0	2	0

Figure 19: **Action Atlas: Feature Explorer.** UMAP (McInnes et al., 2018) projection of 4,096 SAE features from OpenVLA-OFT layer 16, colored by semantic category.



Enter query

Select a concept to see related features across layers

Compare baseline policy with SAE reconstruction (encode → decode). Both should succeed.

Task 3: Original policy behavior

Task 3: Per-token SAE encode → decode

Key Finding

Per-token SAE reconstruction preserves task behavior. Both baseline and SAE reconstruction successfully complete the task, demonstrating that the SAE captures essential action information without losing critical behavioral details. This validates our per-token SAE approach.

Enter prompt...

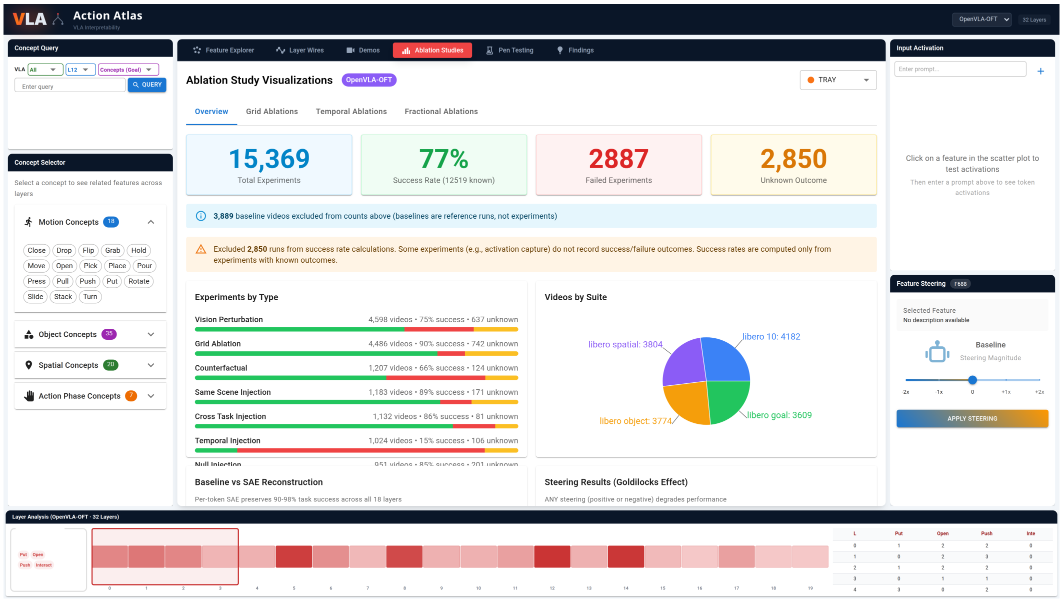
Click on a feature in the scatter plot to test activations

Then enter a prompt above to see token activations

Selected Feature
No description available

L	Put	Open	Push	Inte
0	1	2	2	0
1	0	2	3	0
2	1	2	2	0
3	0	1	1	0
4	3	0	2	0

Figure 20: **Action Atlas: Video Library.** Multi-filter interface for browsing 10,000+ rollout videos.



Ablation Study Visualizations

15,369 Total Experiments

77% Success Rate (12519 known)

2887 Failed Experiments

2,850 Unknown Outcome

3,889 baseline videos excluded from counts above (baselines are reference runs, not experiments)

Excluded 2,850 runs from success rate calculations. Some experiments (e.g., activation capture) do not record success/failure outcomes. Success rates are computed only from experiments with known outcomes.

Select a concept to see related features across layers

Click on a feature in the scatter plot to test activations

Then enter a prompt above to see token activations

Selected Feature
No description available

Experiments by Type

- Vision Perturbation 4,598 videos • 75% success • 637 unknown
- Grid Ablation 4,486 videos • 90% success • 742 unknown
- Counterfactual 1,207 videos • 66% success • 124 unknown
- Same Scene Injection 1,183 videos • 89% success • 171 unknown
- Cross Task Injection 1,132 videos • 86% success • 81 unknown
- Temporal Injection 1,024 videos • 15% success • 106 unknown
- Null Injection 951 videos • 85% success • 201 unknown

Videos by Suite

Baseline vs SAE Reconstruction

Per-token SAE preserves 90-98% task success across all 18 layers

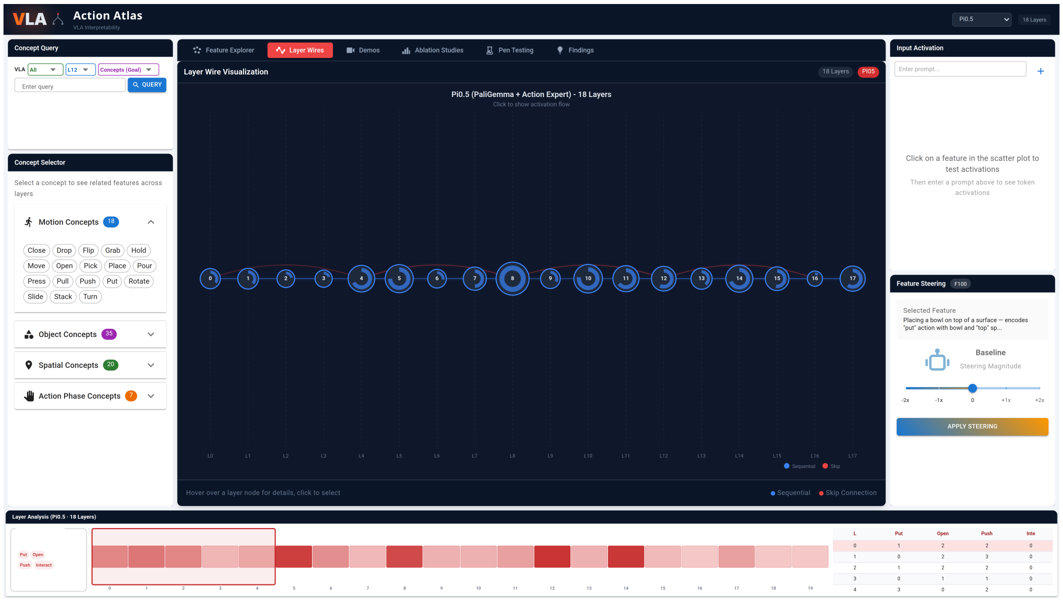
Steering Results (Goldilocks Effect)

ANY steering (positive or negative) degrades performance

Layer Analysis (OpenVLA-OFT · 32 Layers)

L	Put	Open	Push	Inte
0	1	2	2	0
1	0	2	3	0
2	1	2	2	0
3	0	1	1	0
4	3	0	2	0

Figure 21: **Action Atlas: Ablation Comparison.** Three-column video comparison showing baseline, SAE-reconstructed, and ablated rollouts side-by-side.



Layer Wire Visualization

Pi0.5 (PaliGemma + Action Expert) - 18 Layers

Click to show activation flow

Hover over a layer node for details, click to select

● Sequential ● Skip Connection

Select a concept to see related features across layers

Click on a feature in the scatter plot to test activations

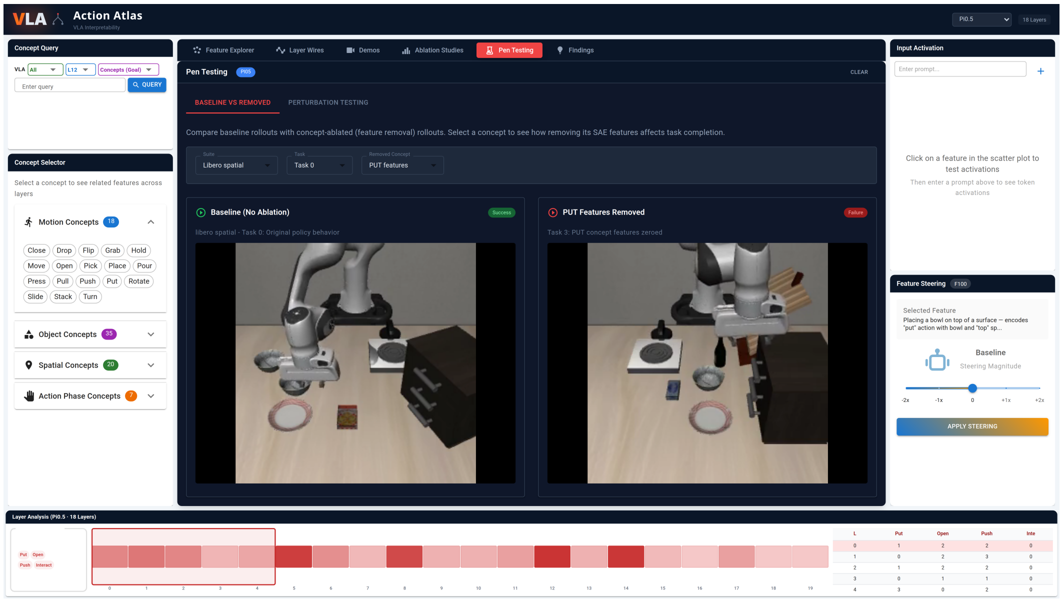
Then enter a prompt above to see token activations

Selected Feature
Placing a bowl on top of a surface — encodes "put" action with bowl and "top" sp...

Layer Analysis (Pi0.5 · 18 Layers)

L	Put	Open	Push	Inte
0	1	2	2	0
1	0	2	3	0
2	1	2	2	0
3	0	1	1	0
4	3	0	2	0

Figure 22: **Action Atlas: Layer Wires.** Interactive visualization of information flow across transformer layers.



Compare baseline rollouts with concept-ablated (feature removal) rollouts. Select a concept to see how removing its SAE features affects task completion.

Suite: Libero spatial

Task: Task 0

Removed Concept: PUT features

Select a concept to see related features across layers

libero spatial - Task 0: Original policy behavior

Task 3: PUT concept features zeroed

Click on a feature in the scatter plot to test activations

Then enter a prompt above to see token activations

Selected Feature

Placing a bowl on top of a surface — encodes "put" action with bowl and "top" sp...

L	Put	Open	Push	Inte
0	1	2	2	0
1	0	2	3	0
2	1	2	2	0
3	0	1	1	0
4	3	0	2	0

Figure 23: **Action Atlas: Perturbation Testing.** Interface for applying real-time visual perturbations and observing behavioral effects.