Not All Features Are Created Equal: A Mechanistic Study of Vision-Language-Action Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

A fine-tuned Vision-Language-Action (VLA) policy will pick up the alphabet soup and place it in the basket on demand, then drop the soup off the table when an evaluator shifts the basket five centimeters left. We use activation injection to ask what the policy is doing: inject task A's activations into task B's scene at the action expert, and $\pi_{0.5}$ executes task A's motor trajectory in 99.6% of episodes (n=1,968); X-VLA does so in 99.8%. The injected program is bound to absolute workspace coordinates rather than the visible scene, which mechanistically explains the perturbation brittleness reported in concurrent benchmark work. The same intervention framework applied across six VLAs ($\pi_{0.5}$, OpenVLA-OFT, X-VLA, SmolVLA, GR00T N1.5, and ACT as a language-free control) on LIBERO, MetaWorld, SimplerEnv, and ALOHA over 420,000+ rollouts surfaces three further findings: language is encoded by every architecture yet behaviorally ignored when vision uniquely identifies the goal; SAE pooling preference splits along architecture lines ($\pi_{0.5}$ per-token, X-VLA mean-pool, SmolVLA indifferent); and pathway specialization replicates wherever expert and VLM pathways are separable. SmolVLA's interleaved fusion attenuates the headline to 52.1% LIBERO override, scoping universality. We release **Action Atlas** (https://action-atlas.com[1]), an interactive feature-exploration platform covering all 520+ trained SAEs and 79 identified concepts. Anonymized repo: https://beebo692.github.io/nips12/.

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 follow language instructions and identify *what* to manipulate, or do they replay coordinate-bound motor programs that hit *where* the training distribution put the object?

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 whose kinematic and control models are inspectable by construction [Craig, 2018].

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, polysemantic

[1] Site identifying content anonymized for blind review.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

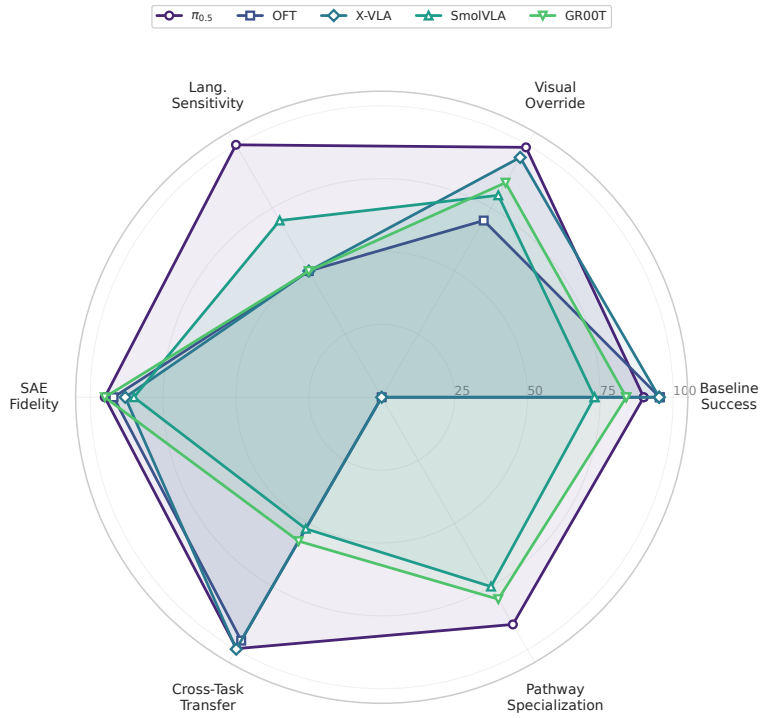


Figure 1: Cross-model capability radar across the five language-capable VLAs. Axes: baseline success, visual override, language sensitivity (inverted: 100 = insensitive), SAE fidelity, cross-task transfer, pathway specialization.

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-relevant 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.

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 (**420,000+ rollout episodes**). Cross-task activation injection redirects 99.6–99.8% of $\pi_{0.5}$ and X-VLA episodes onto source-task trajectories despite the destination scene, demonstrating that fine-tuned VLAs run motor programs bound to absolute workspace coordinates, not abstract task representations; this binding mechanistically explains the brittleness pattern of Zhou et al. [2025]. Three further findings replicate across architectures. (1) Language sensitivity is task-structure-dependent, not architecture-dependent: when vision uniquely identifies the goal, prompts are behaviorally ignored ($\pi_{0.5}$ ANOVA p=0.247, η^2=0.0015) despite linear probes classifying them at 99.3% (chance $1/N$). (2) SAE pooling preference is architecture-tied, not universal: $\pi_{0.5}$ requires per-token (per-token 81% / mean-pool 3%), X-VLA prefers mean-pool (51/81% per-token/mean-pool over the full 24-layer stack), and SmolVLA shows no architecturally-decisive preference (per-token 67.2% vs. mean-pool 66.4%, Δ=0.8pp grand-mean across all 4 LIBERO suites at full 32-layer scale; per-suite $|\Delta| \leq 3.1$pp, sign flips on `libero_10` and `spatial`); the temporal-contrastive arm collapses adjacent latent codes ($\cos_{\text{adj}}$:0.86$\rightarrow$0.997) and degrades all three architectures. (3) Pathway

specialization replicates across all three multi-pathway architectures ($\pi_{0.5}$, SmolVLA, GR00T): expert pathways encode motor programs, VLM pathways encode goals. SmolVLA's interleaved VLM-expert fusion attenuates cross-task override to 52.1% on LIBERO (n=720, expert pathway 78.6%, VLM pathway 25.6%) and 12.4% on MetaWorld.

Our contributions include:

- Cross-architecture mechanistic analysis at scale: the first systematic study spanning six architectures (80M–7B, four benchmarks, 420,000+ episodes). The four findings above replicate across architectures.
- Causal evidence for coordinate-bound motor programs: cross-task injection (§4.3) and pathway specialization (§4.5) jointly localize the substrate, mechanistically explaining the brittleness pattern of Zhou et al. [2025].
- Architecture-tied SAE pooling preference: a three-arm comparison (per-token, mean-pool, temporal-contrastive) on $\pi_{0.5}$, X-VLA, and SmolVLA shows that pooling preference is not universal: $\pi_{0.5}$ requires per-token, X-VLA prefers mean-pool, and SmolVLA is indifferent at the full-stack scale. The temporal arm collapses adjacent latent codes ($\cos_{\text{adj}}$:0.86 $\rightarrow$ 0.997) and degrades all three architectures.
- Action Atlas, an open-source interactive feature-exploration platform (https://action-atlas.com[2]) covering all 520+ SAEs, 79 identified concepts, 10,000+ rollout videos, and ablation comparisons across all six models.

2 Related Work

VLA models extend vision-language pretraining to robotic control along several distinct design axes. RT-2 [Zitkovich et al., 2023] demonstrated that VLMs can generate tokenized robot actions, establishing the discrete-token paradigm; building on this, OpenVLA-OFT [Kim et al., 2025] replaced discrete tokenization with continuous L1 regression and achieved 97.1% LIBERO success. A parallel line of work moved away from autoregressive decoding entirely: $\pi_{0.5}$ [Physical Intelligence, 2025] introduced flow matching with a dedicated action expert, sidestepping bin-collapse failures of discrete tokens. The six models we study (Table 1) span this design space, from 80M (ACT) to 7B (OFT) parameters. Independent robustness evaluations [Fei et al., 2025, Zhou et al., 2025] have shown a pattern that our interpretability study explains mechanistically: small object-position perturbations drop OpenVLA and π_0 to 0.00 success across all four LIBERO suites and $\pi_{0.5}$ to 0.08–0.38 (vs. 0.92–0.98 on the unperturbed originals) [Zhou et al., 2025], while instruction-level perturbations drop all three to 0.00 on most suites despite passing semantic paraphrases. Together these failures are consistent with memorization-based rather than language-grounded behavior.

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]. Concurrent VLA-interpretability work is single-architecture: Häon et al. [2025] steer π_0/OpenVLA SAE features, Molinari et al. [2025] probe world-model structure, Khan et al. [2025] train SAEs on Magma. None perform cross-task injection or compare pooling strategies; our six-architecture design covers both, plus pathway-split analysis. Extended related work in App. A.

3 Method

We apply four techniques uniformly across all six models: activation injection, counterfactual prompting, sparse autoencoders (SAEs), and linear probes. Figure 2 illustrates the pipeline.

3.1 VLA Architectures Under Study

The six architectures span 80M to 7B parameters and four action-generation paradigms: flow matching [Lipman et al., 2023], continuous regression, diffusion, and CVAE decoding (Tab. 1).

[2]Site identifying content partially anonymized for blind review; full attribution restored at camera-ready.

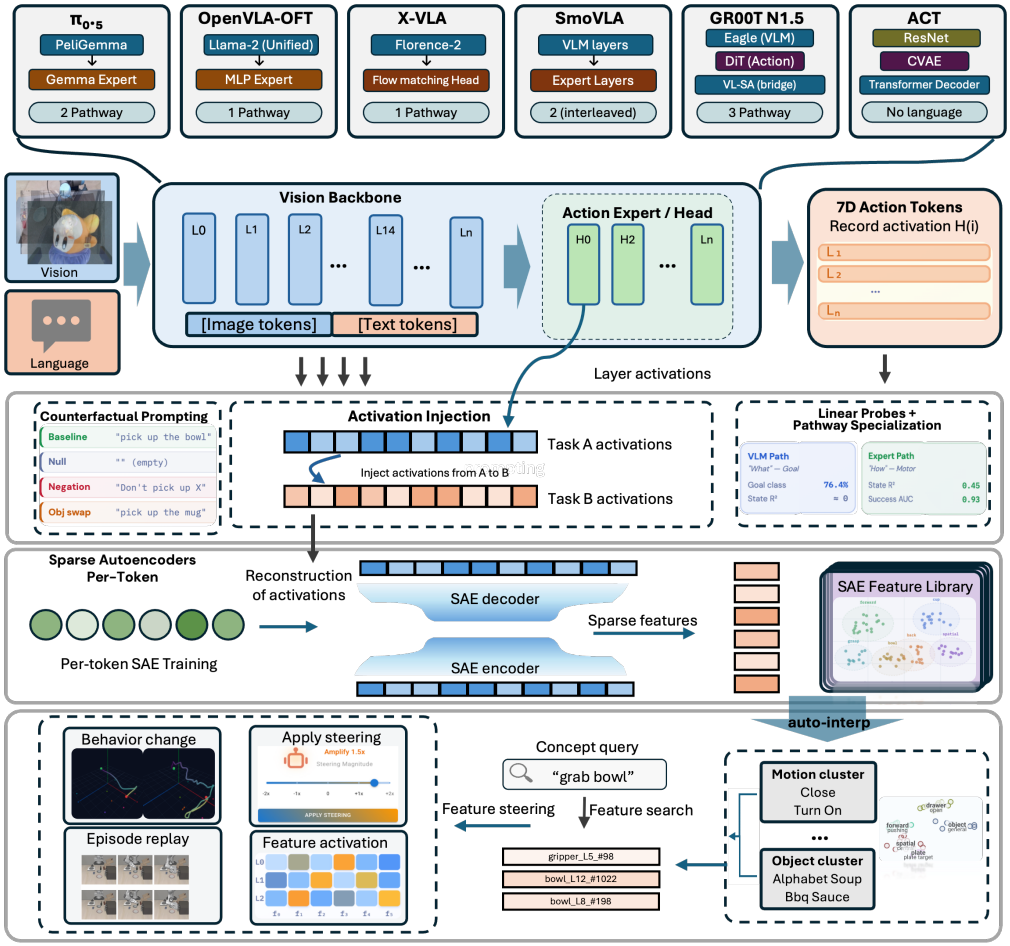


Figure 2: Methodology overview. Top: record VLA activations during rollouts, replay under counterfactual conditions, read causality from behavioral change. Middle: per-token SAEs decompose activations. Bottom: cluster, search, and causally validate features (browsable via Action Atlas).

Table 1: Architectures under study. PG = PaliGemma, Exp. = Expert, Flor. = Florence, D+E+V = DiT + Eagle + VL-SA, LIB. = LIBERO, MW = MetaWorld, SimpE. = SimplerEnv. Details in App. K.1.

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]	∼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

3.2 Activation Injection

Activation injection extends activation patching [Meng et al., 2022] to full rollout episodes: we replace 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 A, then at every timestep t of B override the residual stream at the chosen layers,

$$\mathbf{H}_t^{B,(\ell)} \leftarrow \mathbf{H}_t^{A,(\ell)}, \qquad \ell \in \mathcal{L}_{\text{inject}}, \tag{1}$$

and let the rest of the forward pass proceed normally. The injection set $\mathcal{L}_{\text{inject}}$ ranges from a single layer (e.g., $\{0\}$ for the layer-0 sufficiency test) to all layers. When prompts A and B produce residual streams of different sequence lengths, we overwrite only the 50 action-token positions, which are

fixed across prompt configurations; image and prompt tokens pass through unchanged. When the target runs longer than the recorded source ($T_B > T_A$), injection ceases at $t{=}T_A$ and the target policy continues unmodified (<5% of cross-task episodes; restricting to $T_B \leq T_A$ shifts override $\leq$1pp on $\pi_{0.5}$ `libero_goal`).

We test three 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 scenes. For multi-pathway models ($\pi_{0.5}$, SmolVLA, GR00T), we also inject into individual pathways to isolate their contributions.

3.3 Counterfactual Prompting

We 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 $m = 8d$. Whether per-token preservation matters depends on architecture (§4.6): on $\pi_{0.5}$, mean-pooling across the 50 action tokens collapses approach, manipulation, and terminal phases into one vector and drops rollout success from 94% to 3% despite normalized MSE ≤ 0.05 ($R^2 > 0.95$). We follow Gao et al. [2024] and report normalized MSE (not "loss recovered"); per-token processing is the conservative default, with per-architecture pooling effects in §4.6 and Appendix G.2. Concept-specific features are identified via 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. Across all six models we train **520+ distinct SAEs**: $\pi_{0.5}$ 72 (18 layers $\times$ 2 pathways $\times$ 2 pooling), OFT 96 (32 layers $\times$ 3 pooling arms), X-VLA 120 (24 layers $\times$ 2 pooling $\times$ 2 environments + temporal), SmolVLA 146 (32 layers $\times$ 2 pathways $\times$ 2 pooling + 18 temporal), GR00T 87 (32 layers $\times$ 2 pooling + 23 temporal); these collectively identify **79 unique manipulation concepts** across motion, object, and spatial categories.

3.5 Linear Probes for Action Prediction

Linear probes [Alain and Bengio, 2017] test whether action information is linearly decodable from intermediate representations. We train ridge regression probes per action dimension and verify causality by projecting out the probe direction.

3.6 Metrics

We evaluate three primary metrics. **Action Cosine Similarity** measures behavioral alignment between episodes, $\cos(\mathbf{a}^X, \mathbf{a}^Y) = \langle \mathbf{a}^X, \mathbf{a}^Y \rangle / (\|\mathbf{a}^X\| \|\mathbf{a}^Y\|)$, computed over the full per-token action vectors. **Task Success** is a binary indicator determined by the environment's built-in success criteria; we report change as $\Delta\text{SR} = \text{SR}_{\text{intervention}} - \text{SR}_{\text{baseline}}$. **Override Rate** quantifies how often the injected behavior dominates the prompted behavior in cross-task injection. Let $\Delta_t = \cos(\mathbf{a}_t^{B|A\text{-inj}}, \mathbf{a}_t^A) - \cos(\mathbf{a}_t^{B|A\text{-inj}}, \mathbf{a}_t^B)$; we define the override rate conditional on exclusion of a cosine-tie band $|\Delta_t| \leq \tau$,

$$\text{OR} \;=\; \Pr[\, \Delta_t > \tau \mid |\Delta_t| > \tau \,], \tag{2}$$

with $\tau = 0.05$. Tie-band fractions vary by architecture ($\pi_{0.5}$ 0.3%, OFT 20.9%, X-VLA 60.0%, SmolVLA 0.0%) and are reported with the X-VLA tie-inclusive override in Tab. 4. All reported confidence intervals are 95% Wilson score intervals; ANOVA effect sizes are reported as η^2.

4 Experiments

We evaluate 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), X-VLA [Zheng et al., 2026] (1B, soft-prompted flow), SmolVLA [Shukor et al., 2025] (450M, interleaved VLM-expert), GR00T N1.5 [Bjorck et al., 2025] (3B, DiT-Eagle-VL-SA), and ACT [Zhao et al., 2023] (80M, CVAE, vision-only). We collect **420,000+ rollout episodes** across 12 experiment types, 4 benchmarks, and up to 50 tasks per environment, on 8×A100-80GB, RTX 5090, and 2×RTX 4090. Visual context selects the action (§4.2); the action is a coordinate-bound motor program (§4.3); language matters only when vision is ambiguous (§4.4); pathway specialization implements the bind/select interface (§4.5); SAEs expose feature-level redundancy and tie the mechanism to the brittleness of Fei et al. [2025], Zhou et al. [2025] (§4.6).

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.

Table 2: Experimental scale across six models. Episodes aggregate across all applicable experiment types; not every model runs every type. SAE counts include per-token, mean-pool, and temporal variants. Per-model concept counts overlap; the total row reports unique concepts after deduplication.

Model	Episodes	SAEs	Concepts
$\pi_{0.5}$	71,500+	72	43
OpenVLA-OFT	70,700+	96	45
X-VLA	50,000+	120	82
SmolVLA	58,000+	146	45
GR00T N1.5	164,700+	87	36
ACT	1,870	–	–
Total	**>420,000**	**520+**	**79**

4.2 Visual Context Dominates Action Generation

Visual context dominates action generation across all six architectures (Tab. 3). Injecting baseline visual-pathway activations under a null prompt restores cosine similarity to ≥ 0.99 and task success to 14–77%; zeroing the visual pathway collapses success to the noise floor on every model. Recovery is largest for dual-pathway designs ($\pi_{0.5}$ 73–77%) and smallest for single-pathway OFT (14–15%, n=120 per layer); the gap reflects visual-language entanglement absorbing more of the prompt-conditioning load in single-stack architectures (§4.4). Per-model layer-zeroing profiles in App. G.

Within a shared scene, same-scene injection of prompt-B activations into a prompt-A rollout steers behavior toward the injected target on every multi-pathway model tested. Layer-zeroing sensitivity is architecture-tied (every layer required on X-VLA; $\pi_{0.5}$/SmolVLA tolerate late-layer ablation); per-model profiles in Appendix G.

4.3 Motor Programs Bind to Absolute Spatial Coordinates

Does the injected activation encode a task concept ("pick up the cup") or a coordinate-bound motor program? Cross-task injection separates the two. Across all five language-capable VLAs, injection collapses destination-task success, yet trajectory analysis shows the injected activations steer behavior toward source-task positions (Tab. 4; ACT serves as the language-free control). Override rate (Eq. 2) is the conditional probability that source cosine exceeds destination cosine by more than the tie-band threshold τ=0.05.

Displacement analysis (Figure 3) resolves an apparent contradiction: injection "fails" task success but "succeeds" at steering behavior. $\pi_{0.5}$, X-VLA, and OFT show strong source-dominant trajectories ($\geq 96\%$ override); GR00T moderate (57.0%); SmolVLA's expert pathway overrides at $3.07\times$ the

Table 3: Visual-pathway influence per suite. Baselines from a unified 25-ep×10-task batch (n=250/suite); OFT null prompt at n=50/suite. The OFT null suite-spread (12–96%) is itself evidence for suite-dependent language sensitivity (§4.4). Intervention rows: same-scene inject = baseline activations into same-task episode; zero any layer = mean across the layer sweep. †GR00T `libero_spatial`: public N1.5 spatial checkpoint underperforms the reference (App. J). Per-experiment n in App. G.

Model	Condition	Goal	Object	Spatial	Long / 10
$\pi_{0.5}$	Baseline	93.2%	87.6%	92.8%	86.0%
	Null prompt, no injection	0.0%	0.0%	0.0%	0.0%
	Null + PaliGemma-ALL inject	83.3%	66.7%	100.0%	—
	Same-scene inject ALL	91.0%	87.5%	88.2%	75.2%
OFT	Baseline	98.8%	99.2%	92.4%	91.2%
	Null prompt, no injection	12.0%	96.0%	58.0%	—
	Null + inject zero-baseline (rec.)	14.1%	14.6%	14.4%	13.5%
X-VLA	Baseline	98.8%	98.8%	95.6%	87.6%
	Null prompt, no injection	10.0%	60.0%	48.0%	28.0%
	Zero any single layer	0.0%	0.0%	0.0%	0.0%
SmolVLA	Baseline	74.8%	87.6%	65.6%	41.2%
	Zero expert layer (mean)	71.0%	76.5%	61.8%	19.2%
GR00T N1.5	Baseline	93.6%	95.6%	69.2%†	77.2%
	Null source (inject ALL DiT)	3.3%	3.3%	—†	3.3%
	Zero any DiT layer (L0–L15)	0.0%	0.0%	—†	0.0%
ACT (ALOHA)	Baseline	100.0% (both ALOHA tasks)			
	Mask workspace grid (2,2)	10.0% (both ALOHA tasks)			

Table 4: Cross-task injection across six models. Dst. SR: destination-task success under injection. Override rate (Eq. 2): tie-band-conditional at τ=0.05; tie-inclusive variants and per-architecture tie fractions in App. B. ACT injection is a no-op ($\cos$ =1.0); its 70.0% Dst. SR is the Insertion-baseline rate on the 10 cross-task pairs (different from Tab. 3's TransferCube baseline).

Model	Pairs (n)	Dst. SR	Override rate	Notes
$\pi_{0.5}$	1,968	2.6%	99.6%	goal/spatial/object; 95% Wilson CI ±0.4pp
OFT	1,079	5.0%	96.4%	90.8–100.0% per suite (Tab. 11)
X-VLA	3,150	0.0%‡	99.8%	‡LIBERO; WidowX 54.2% (task overlap)
SmolVLA	720	0.0%	52.1%§	§LIBERO aggregate; expert pathway 78.6%, VLM pathway 25.6%
GR00T	270	0.0%	57.0%¶	¶goal 85.6%, object 52.2%, long 33.3%
ACT (ctrl.)	10	70.0%‖	0.0%‖	‖injection is a no-op ($\cos$ =1.0); 70.0% is destination-baseline SR

VLM-pathway rate (78.6% vs. 25.6%, LIBERO; aggregate 52.1%, n=720). Under injection 68–80% of episodes displace an object (>5 mm), but the moved object belongs to the destination scene; DTW on EEF trajectories yields 86–91% source-dominance on `libero_goal`/`libero_spatial` (App. G.5). A norm-matched random-direction control yields 22% source-behavior (n=50, App. C.3); the 99.6% override is direction-specific, not magnitude-driven. The robot reaches toward where source objects would have been, executing spatially grounded motor programs bound to absolute workspace coordinates, consistent with Fei et al. [2025], Zhou et al. [2025].

4.4 Language Becomes Necessary Only When Vision is Ambiguous

If visual context selects the action, language should matter only when the scene fails to disambiguate the task. Counterfactual prompting across 11,226 episodes (five language-capable VLAs) confirms this: sensitivity tracks task structure, not architecture. On the four LIBERO suites and four MetaWorld difficulty bands, visually unambiguous tasks (`libero_object`, MetaWorld easy) retain 60–100%

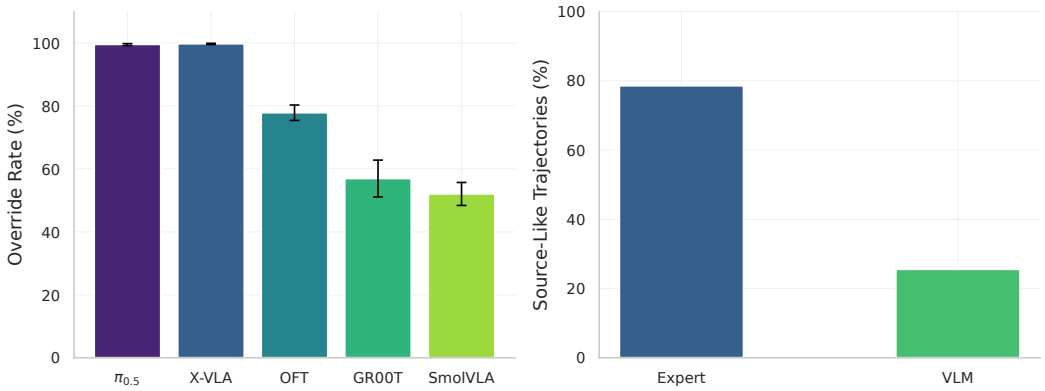


Figure 3: **Cross-task displacement override rates.** Left: override rate across five models. $\pi_{0.5}$ (99.6%, n=1,968) and X-VLA (99.8%, n=3,150) show near-complete source behavior transfer; OFT 96.4% (n=1,079); GR00T 57.0% (n=270, suite-dependent: `goal` 85.6%, `long` 33.3%). Error bars: 95% Wilson CIs. Right: SmolVLA pathway displacement on LIBERO (78.6% expert vs. 25.6% VLM, n=720).

success under null and wrong-prompt conditions on every architecture, while visually ambiguous tasks (`libero_goal`, MetaWorld hard) collapse to 0–10% on the same architectures. $\pi_{0.5}$ ANOVA on `libero_object` yields $F(4, 3391)$=1.23, p=0.247, η^2=0.0015 (n=3,396, MDE at 80% power η^2=0.0035, so any true effect is below Cohen's small); per-model ANOVA appears in Appendix G.

Table 5: Counterfactual prompting across models (total n=11,226; per-model n in App. G). ‡Correct-prompt SR is the matched within-experiment control; differs from Tab. 3 baselines by 5–13 pp due to independent batches. §Object-swap not run on MetaWorld.

Model / Suite	Correct Prompt‡	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	100.0%	100.0%	100.0%
OFT / goal	100.0%	10.0%	10.0%
OFT / spatial	90.0%	70.0%	60.0%
X-VLA / object	100.0%	60.0%	60.0–90.0%
X-VLA / goal	94.0%	10.0%	4.0–10.0%
X-VLA / spatial	98.0%	48.0%	44.0–58.0%
SmolVLA / MW easy	85.0%	82.0%	—
SmolVLA / MW hard	62.0%	41.0%	—
GR00T / object	93.0%	70.0%	50.0%
GR00T / goal	97.0%	0.0%	0.0%
GR00T / long	83.0%	67.0%	47.0%

Despite behavioral invariance, linear classifiers recover prompt category from late-stack activations at >99% accuracy on every language-capable architecture (Tab. 15): prompts are encoded but not used. Table 5 shows the suite-dependent pattern replicates across architectures: visually unambiguous suites stay prompt-immune (60–100% under wrong prompts) while visually ambiguous suites collapse to 0–10%. The common factor is whether visual context alone identifies the target, not model design.

4.5 Expert Pathways Encode HOW; VLM Pathways Encode WHAT

The bind/select division of labor predicts a corresponding architectural division: VLM pathways encode WHAT (goal selection); motor-expert pathways encode HOW (coordinate-bound motor programs). All three multi-pathway models show this dissociation under different fusion designs. Injecting expert pathways causes active misdirection (reaching to wrong locations); injecting VLM pathways causes passive stalling (timeout). On SmolVLA, oracle state-prediction probes recover ground-truth state from expert activations at 4.5× the VLM rate (oracle ratio 0.58 vs. 0.13, Tab. 15). $\pi_{0.5}$ reproduces the same direction qualitatively: PaliGemma goal classification rises to 76.4% while

expert goal classification settles at 62.6% (App. F.1); GR00T DiT layers are more ablation-sensitive (40–80% drop) than VL-SA layers (Fig. 11), consistent with the same expert-encodes-state pattern under a different fusion. The split is invariant to fusion strategy and supports runtime failure diagnosis: misdirection signals expert failure; stalling signals VLM failure.

4.6 SAE Layer Profile and Pooling Sensitivity

To read the feature content of expert pathways (§4.5), we train per-token TopK SAEs at each pathway layer (Eqs. 3–5; 520+ SAEs) and find two cross-architecture patterns.

Single-feature ablation is largely tolerated by the VLM backbones (28–92% zero-effect rates; Fig. 7), consistent with concepts encoded across multiple features, not localized in one. Sensitivity is layer-localized rather than diffuse and the action pathway behaves differently from the language stack (Fig. 4): the $\pi_{0.5}$ expert (18 layers) and GR00T DiT (16 layers) kill the policy on $\geq$40% of ablation pairs across most of their depth, while OFT (32 L), X-VLA (24 L), and SmolVLA expert (32 L) keep the kill-switch rate below 10% almost everywhere. The per-layer SAE rollout in Fig. 5 replicates the same depth profile under per-token reconstruction. Width does not predict sensitivity (SmolVLA 480-dim 28%, X-VLA 1024-dim 82%, OFT 4096-dim 92%); per-model profiles in Appendix G.

A five-architecture three-arm rollout comparison shows two pooling regimes (Tab. 8). Per-token-required: $\pi_{0.5}$ 81/3%, OFT 80/7% (mean-pool collapse). Pooling-tolerant: X-VLA 51/81% (mean-pool preferred), SmolVLA 67.2/66.4% (indifferent), GR00T-DiT 80/83% (mean-pool preferred). The temporal-contrastive arm collapses adjacent codes ($\cos_{\text{adj}} : 0.86 \rightarrow 0.997$) at R^2=0.97 on every architecture; this collapse drives the degradation independent of pooling preference.

4.7 Limitations

All experiments run in simulation; physical replication on UR10e/Franka is in progress. Cross-task injection establishes that the action expert can override visual context, but does not separate workspace-frame from scene-relative binding; the rigid-transform disambiguation control is in App. B. GR00T public N1.5 spatial underperforms the published reference, so spatial intervention coverage is limited († in Tab. 3, App. J). X-VLA (99.8%) and GR00T (57.0%) override use action-vector cosine; an end-effector-velocity classifier yields lower numbers (52.8%, 6.7%) and we report the original for cross-model comparability. Extended limitations (counterfactual coverage, steering sensitivity, injection confounds) appear in Appendix B.

5 Conclusion

Across six architectures and 420,000+ episodes, fine-tuned VLAs encode coordinate-bound motor programs that the visual pathway selects from a small library. Cross-task activation injection redirects 99.6% of $\pi_{0.5}$ and 99.8% of X-VLA episodes onto source-task trajectories despite the destination scene, and language is behaviorally ignored when vision uniquely identifies the goal ($\pi_{0.5}$ ANOVA p=0.247, η^2=0.0015). Wherever a model exposes separable visual and expert pathways, the visual side encodes what to do and the expert side encodes how to do it, a dissociation that replicates across the multi-pathway architectures we studied. The same binding accounts for the perturbation brittleness reported in concurrent benchmark work [Zhou et al., 2025, Fei et al., 2025], because translating an object out of its training-distribution position decouples it from the bound coordinates and leaves the policy heading toward the prior workspace location. SAE pooling preference follows the same architectural divide. $\pi_{0.5}$ and OFT carry phase-distinct information in the residual stream and need per-token reconstruction, while X-VLA, SmolVLA, and GR00T-DiT carry approximately position-invariant action tokens and tolerate or prefer mean-pool. We release 520+ trained SAEs and an interactive feature-exploration platform (Action Atlas), and an in-progress UR10e/Franka physical replication will test whether the binding survives outside simulation.

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.

Usha Bhalla, Alex Oesterling, Claudio Mayrink Verdun, Himabindu Lakkaraju, and Flavio P. Calmon. Temporal sparse autoencoders: Leveraging the sequential nature of language for interpretability. *arXiv preprint arXiv:2511.05541*, 2025. URL https://arxiv.org/abs/2511.05541. ICLR 2026 Oral.

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.

Jiayi Chen, Wenxuan Song, Shuai Chen, Jingbo Wang, Zhijun Li, and Haoang Li. DFM-VLA: Iterative action refinement for robot manipulation via discrete flow matching. *arXiv preprint arXiv:2603.26320*, 2026. URL https://arxiv.org/abs/2603.26320.

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.

John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson, Hoboken, NJ, 4 edition, 2018. ISBN 9780133489798.

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.

Davide Ghilardi, Federico Belotti, Marco Molinari, Tao Ma, and Matteo Palmonari. Group-SAE: Efficient training of sparse autoencoders for large language models via layer groups. *arXiv preprint arXiv:2410.21508*, 2024. URL https://arxiv.org/abs/2410.21508.

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*, pages 2743–2762. PMLR, 2025. URL https://arxiv.org/abs/2509.00328.

Yutong Hu, Jan-Nico Zaech, Nikolay Nikolov, Yuanqi Yao, Sombit Dey, Giuliano Albanese, Renaud Detry, Luc Van Gool, and Danda Paudel. AR-VLA: True autoregressive action expert for vision–language–action models. *arXiv preprint arXiv:2603.10126*, 2026. URL https://arxiv.org/abs/2603.10126.

Yuhua Jiang, Shuang Cheng, Yan Ding, Feifei Gao, and Biqing Qi. AsyncVLA: Asynchronous flow matching for vision–language–action models. *arXiv preprint arXiv:2511.14148*, 2025. URL https://arxiv.org/abs/2511.14148.

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.

Chiraag Kaushik, Davis Barch, and Andrea Fanelli. Decomposing multimodal embedding spaces with group-sparse autoencoders. *arXiv preprint arXiv:2601.20028*, 2026. URL https://arxiv.org/abs/2601.20028.

M. A. Khan, N. Boskov, F. M. Anwar, and M. A. Khan. 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.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. *arXiv preprint arXiv:2604.13645*, 2026. URL https://arxiv.org/abs/2604.13645.

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, Ishika Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. 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)*, pages 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.

Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, and Mengzhen Liu. HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025a. URL https://arxiv.org/abs/2503.10631.

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*, 2025b. URL https://arxiv.org/abs/2507.01424.

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, Leonardo Nevali, Saharsha Navani, and Omar G. Younis. 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.

Ali Nasiri-Sarvi, Hassan Rivaz, and Mahdi S. Hosseini. SPARC: Concept-aligned sparse autoencoders for cross-model and cross-modal interpretability. *arXiv preprint arXiv:2507.06265*, 2025. URL https://arxiv.org/abs/2507.06265.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, 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, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. 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. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. URL https://arxiv.org/abs/2604.15483.

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*, pages 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)*, pages 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)*, pages 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.

Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G. Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 59304–59325, 2025. URL https://arxiv.org/abs/2502.03714.

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*, pages 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, $\pi_{0.5}$, $\pi^*_{0.6}$, and $\pi_{0.7}$ models [Black et al., 2024, Physical Intelligence, 2025, Physical Intelligence et al., 2025, Physical Intelligence, 2026] introduced flow matching with a dedicated action expert and progressively more diverse training context; whether $\pi_{0.7}$'s reported emergent capabilities reflect genuine compositional generalization rather than further-scaled memorized motor patterns is an open question that the mechanistic tools developed here can answer. 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]. Concurrent representation-analysis work in robot policies uses different tools: Lei et al. [2026] apply UMAP and Wasserstein-distance analysis to characterize sim-real co-training, identifying "structured representation alignment" and "importance reweighting" effects; our SAE-based feature-level analysis is complementary.

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. 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 and 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 and Abreu, 2024], SAE-guided additions [Soo et al., 2025], and VLM safety steering [Wu et al., 2025, Sivakumar et al., 2025]. Linear probing [Alain and 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. [2025] introduced VLA steering via internal representations. Molinari et al. [2025] probed for emergent world models. 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., 2025b], dVLA [Wen et al., 2025], and Embodied-R1 [Yuan et al., 2025].

Cross-Model and Multimodal SAE Alignment. Per-model SAE training, including ours, leaves features in incompatible per-architecture spaces, which complicates direct comparison of concepts across pathways or models. Two recent lines of work address this. The first scales SAEs across model

dimensions: Universal Sparse Autoencoders [Thasarathan et al., 2025] and SPARC [Nasiri-Sarvi et al., 2025] learn a single shared dictionary across multiple networks, enforcing concept alignment via shared encoders or a global TopK constraint, while Group-SAE [Ghilardi et al., 2024] reduces training cost by tying nearby layers into a single dictionary using residual-stream similarity. The second targets multimodal alignment: Kaushik et al. [2026] address the "split dictionary" phenomenon (where features fire only for one modality) using cross-modal random masking and group-sparse regularization. Both lines are directly relevant to our setting: the split-dictionary problem mirrors our VLM/expert-pathway split (§4.5), and a multimodal-aligned SAE could test whether the goal/motor dissociation is genuinely orthogonal at the dictionary level or an artifact of independent training. We leave this extension to future work.

Concurrent VLA Architectures. Several recent VLA designs explore alternative action-decoding paradigms whose binding behavior our framework can probe directly. On the autoregressive side, AR-VLA [Hu et al., 2026] treats action generation as a true causal sequence with a long-lived action-history memory, breaking from chunk-based reactive heads. On the flow-matching side, DFM-VLA [Chen et al., 2026] replaces fixed-token decoding with discrete flow-matching iterative refinement, and AsyncVLA [Jiang et al., 2025] introduces asynchronous flow matching with a confidence rater for selective regeneration. Bridging the two paradigms, HybridVLA [Liu et al., 2025a] unifies discrete autoregression and continuous diffusion within a single LLM backbone. Each design changes the pathway through which an action is produced from a visual scene, but the cross-task injection and pathway-zeroing protocols in this paper transfer mechanically; the only architecture-specific concern is the hook placement choice documented in App. D.4. Whether the coordinate-bound binding documented here persists under iterative-refinement (DFM-VLA), confidence-aware regeneration (AsyncVLA), or context-aware autoregression (AR-VLA) is an empirical question that each of these architectures' open releases makes testable. We expect the binding to weaken in proportion to how much an architecture re-grounds on the current observation between action chunks rather than committing to a precomputed plan.

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 grounded motor programs, pathway specialization) persist through domain-specific fine-tuning remains untested. We are extending this study to physical hardware (Universal Robots UR10e and Franka Emika Panda); replication results will be shared on Action Atlas as they land. 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.

Coordinate-Frame Disambiguation

Cross-task injection and the displacement analysis (App. G.5) are consistent with motor programs bound to absolute workspace coordinates, but do not exclude binding to scene-relative coordinates that happen to coincide with the source-task locations. The disambiguation control applies a known rigid transform T to the whole world (objects and robot fixed in their relative configuration) and asks whether injected end-effector trajectories transform by T (workspace-frame binding) or stay invariant in scene-relative coordinates. We do not run this control in the present submission; it is exact in simulation via a single MuJoCo world transform and is the cleanest check planned on the physical setup. Under either workspace-frame or open-loop-replay framings, the policy commits to a trajectory tied to spatial coordinates rather than re-grounding on the visible task, which is the core claim for the brittleness explanation.

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 varies across architectures: $\pi_{0.5}$ expert pathways collapse under suppression (ΔSR $= -84$pp at $-3\times$), GR00T DiT features collapse under amplification (-68pp at $9\times$), while OFT (ΔSR $= -6$pp at $-3\times$) and SmolVLA (-3pp at $5\times$) tolerate the same magnitude of intervention. Architecture and training regime, not VLAs as a class, determine fragility. We attribute the spread to three properties of behavior-cloned manipulation: sub-millimeter end-effector precision requirements [Chi et al., 2023]; reconstruction-error accumulation across the 50-token action chunk, of the same form as autoregressive drift [Bengio et al., 2015]; and the absence of closed-loop correction in behavior-cloned (vs. RL-trained) policies [Ross et al., 2011]. Phase-specific steering on $\pi_{0.5}$ localizes sensitivity to the transport phase ($p = 0.013$, Wilcoxon rank-sum). SAE features are not guaranteed to be disentangled: ablation validation finds that some identified features encode general motor primitives rather than clean semantic concepts.

C Methodology Controls

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 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.

Failure mode (80M, CVAE, vision-only). By construction ACT cannot exhibit language-sensitivity failures, and the visual-pathway dominance result above shows that the vision-only configuration recovers task-relevant behavior without text input, bounding the share of VLA failure attribution that depends on the language modality.

C.3 Random-Direction Injection Control

To rule out the alternative that any large residual perturbation kicks the policy to its closest in-distribution motor prior, we run a norm-matched random-direction control on $\pi_{0.5}$, `libero_goal`, expert layers L16+L17 (the same hook locations as the production cross-task condition). For each forward call t, we replace the source-task injection $\mathbf{h}_\ell^A$ with $r_t \cdot \|\mathbf{h}_\ell^A\|_2$, where $r_t \sim \mathcal{N}(0, I)$ is normalized to the unit sphere. The destination scene and source prompt are unchanged.

Across n=50 episodes (10 pairs $\times$ 5 seeds): **22% source-behavior** [Wilson 95% CI 12.8–35.2], 42% destination-behavior [29.4–55.8], 36% ambiguous, success rate 0%. Mean $\cos_{\text{src}} = -0.017$, mean $\cos_{\text{dst}} = 0.012$, both at noise level. The override rate (cos_src > cos_dst with no threshold) is 50% [36.6–63.4], indistinguishable from chance.

Compared to the directed cross-task condition's 99.6% source-behavior on the same suite/layers (n=1,614, CI [99.2, 99.8]), the gap is 77pp with non-overlapping CIs. The 99.6% override is direction-specific, not magnitude-driven: a Gaussian-random unit direction at the same per-step L2 norm produces neither source nor destination steering, but rather noise that scatters the policy uniformly in residual space.

D Extended Methodology Details

D.1 SAE Architecture and Training

Architecture. Following the TopK design of Gao et al. [2024], our sparse autoencoders consist of an encoder-decoder pair with tied weights. For an input activation $\mathbf{h} \in \mathbb{R}^d$ (per token; $d \in \{480, 1024, 1536, 2048, 4096\}$ across our six models),

$$\mathbf{z} = \mathrm{TopK}_k(\mathbf{W}_e(\mathbf{h} - \mathbf{b}_d)), \tag{3}$$

$$\hat{\mathbf{h}} = \mathbf{W}_e^\top \mathbf{z} + \mathbf{b}_d, \tag{4}$$

where $\mathbf{W}_e \in \mathbb{R}^{m \times d}$ with expansion factor $m \in \{4d, 8d\}$, and $\mathrm{TopK}_k(\mathbf{u})$ retains the $k = 64$ largest pre-activations and zeros the remainder. Tied decoder weights ($\mathbf{W}_d = \mathbf{W}_e^\top$) and the pre-encoder bias subtraction follow Bricken et al. [2023], Gao et al. [2024].

Training objective. TopK enforces hard sparsity, removing the need for an L1 penalty [Cunningham et al., 2023]. We minimize reconstruction MSE plus an auxiliary AuxK term [Gao et al., 2024] that recycles dead features by reconstructing the residual $\mathbf{r} = \mathbf{h} - \hat{\mathbf{h}}$ from the top k_{aux}=512 inactive features:

$$\mathcal{L}(\mathbf{h}) = \underbrace{\|\mathbf{h} - \hat{\mathbf{h}}\|_2^2}_{\text{reconstruction}} + \alpha_{\text{aux}} \underbrace{\|\mathbf{r} - \mathbf{W}_e^\top \mathrm{TopK}_{k_{\text{aux}}}(\mathbf{z}_{\text{dead}})\|_2^2}_{\text{AuxK (dead-feature recycling)}}, \tag{5}$$

with $\alpha_{\text{aux}} = 1/32$. Decoder columns are unit-normalized after each step.

Training Hyperparameters. We train each SAE on 500,000 activation samples (approximately 10,000 forward passes $\times$ 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:

$$\mathbf{h}_{\text{flat}} = \text{reshape}(\mathbf{H}, [B \times 50, 1024]) \tag{6}$$

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:

$$\text{score}_f = d_f \times \text{freq}_f \tag{7}$$

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.

Sample population for Cohen's d. Activations are first max-pooled over the 50 action-token positions per episode, yielding one activation vector per (episode, layer, feature). Cohen's d_f is computed across episodes within each concept set (n=100–200 per concept; standardized mean difference between concept-present and concept-absent episodes). Population statistics use unbiased Welch variance.

Concept specificity scores in Tab. 12(a). The dimensionless score $\text{score}_f = d_f \times \text{freq}_f$ is bounded above by $|d_f|$ and is at most a few units of effect size in our data. The values reported in parentheses in Tab. 12(a) (e.g., 133k, 412k) are not the dimensionless score; they are raw cumulative activation magnitudes per concept, computed as $\sum_{(\text{ep},t)\in\mathcal{C}} \text{ReLU}(z_{f,t,\text{ep}})$ over the concept-positive episode set $\mathcal{C}$ and the 50 action-token positions, summed across layers. We retain the raw-magnitude column for readability of layer rankings within a concept; the per-feature dimensionless score_f used to identify concept-associated features is reported in the supplementary feature-exploration platform.

D.3 Ablation Protocol

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

$$\mathbf{z}_{\text{ablated}} = \mathbf{z} \odot \mathbf{b} \tag{8}$$

$$\mathbf{h}_{\text{modified}} = \mathbf{h} + \mathbf{W}_e^\top(\mathbf{z}_{\text{ablated}} - \mathbf{z}) \tag{9}$$

where $\mathbf{b} \in \{0,1\}^m$ is a binary mask (with m the SAE expansion dimension) carrying zeros at ablated feature indices. Per-token ablation applies this independently to each of the 50 action tokens. By writing the modification as a residual ($\mathbf{W}_e^\top(\mathbf{z}_{\text{ablated}} - \mathbf{z})$) rather than replacing $\mathbf{h}$ with $\hat{\mathbf{h}}_{\text{ablated}}$, we preserve the unmodelled component of the activation and intervene only on what the SAE explains.

Steering operator. The ablation operator generalizes to a continuous steering operator that modulates a target set of features $\mathcal{F}$ by a scalar coefficient α:

$$\mathbf{z}_{\text{steered},f} \;=\; \begin{cases} \alpha\,\mathbf{z}_f & f \in \mathcal{F}, \\ \mathbf{z}_f & \text{otherwise}, \end{cases} \qquad \mathbf{h}_{\text{steered}} \;=\; \mathbf{h} + \mathbf{W}_e^\top(\mathbf{z}_{\text{steered}} - \mathbf{z}). \tag{10}$$

$\alpha = 0$ recovers ablation, $\alpha = 1$ is identity, $\alpha \in (0,1)$ dampens, $\alpha < 0$ negates, and $\alpha > 1$ amplifies. Because TopK enforces hard sparsity, $\alpha\,\mathbf{z}_f = 0$ whenever feature f is inactive in the top-k set; this multiplicative operator therefore modulates only currently-active concepts and cannot inject features absent from the active set. Additive steering ($\mathbf{z}_f \leftarrow \mathbf{z}_f + \alpha\,\mathbf{v}_f$ for a steering vector $\mathbf{v}_f$) would be required to inject inactive concepts and is left to future work. The cross-architecture sensitivity Fig. 7 reports ablation ($\alpha = 0$); we leave a properly-distribution-clipped α sweep to future work.

Cross-task injection as a special case. Equation (1) can be expressed as the SAE-space substitution $\mathbf{z}^B \leftarrow \mathbf{z}^A$, recovering the residual-stream patch used in Section 4.2; ablation and steering are then the diagonal restrictions of this operator to a chosen feature set with coefficient 0 or α respectively.

D.4 Intervention-Hook Protocol

Activation interventions hook into one of three locations per architecture: (i) the full *DecoderLayer output* (residual stream after attention + MLP + residual addition), (ii) the *residual-additive output* of a single sub-block (e.g., MLP output before residual addition), or (iii) the *MLP output* alone. Some VLAs bypass the standard DecoderLayer in their inference path, so hooking the layer wrapper does not modify what the policy consumes.

Per-architecture hook placement. $\pi_{0.5}$, OpenVLA-OFT, X-VLA, and GR00T expose a standard `DecoderLayer.forward` per layer; we hook the layer output (full residual stream) for cross-task injection and zero-layer ablation, matching Eq. (1). SmolVLA's interleaved VLM–expert fusion calls the expert MLP directly inside a hand-written interleaved loop that bypasses the wrapping `DecoderLayer`; hooks installed on the layer output are therefore invisible to the forward pass. We hook `layer.mlp.forward` on SmolVLA, which is the activation written back into the residual stream during inference. X-VLA's soft-prompted Florence-Large path is internally similar; we verified hook visibility via cosine identity ($\cos = 1.0$ between hooked and unhooked baselines on no-op interventions). For SAE reconstruction (Fig. 5, Tab. 7) and SAE feature ablation/steering (Fig. 7, Eq. (9), Eq. (10)), we use MLP-output hooks on every architecture so the SAE operates on the same activation it was trained on.

Reading the layer-zeroing numbers. The OFT *zero any layer* 14–15% recovery in Tab. 3 and the X-VLA / GR00T 0% under the same condition all use the full DecoderLayer-output hook; these are upper bounds on layer importance under standard residual-stream patching. The single-feature ablation kill-switch rates in Fig. 7 use the MLP-output hook. Full-layer-hook zeroing additionally removes the residual passthrough; MLP-only hooks preserve the passthrough and modify only the SAE-explained component.

E Cross-Architecture Synthesis

This section gathers the cross-architecture evidence on a single page so a reader who only wants the cross-model story can stop here; per-architecture deep dives appear in App. J, and the SAE pooling deep dive in App. G.

Five claims, one table or figure each.

1. **Visual-pathway dominance is universal** (Tab. 6, row 1). All six architectures ($\pi_{0.5}$, OFT, X-VLA, SmolVLA, GR00T, ACT) show the same direction under expert-pathway zeroing; the magnitude of the residual recovery is set by architectural pathway separation (e.g., OFT 14% vs. $\pi_{0.5}$ 73%) rather than by whether the phenomenon is present.
2. **Cross-task injection rebinds motor coordinates, not goal semantics** (Tab. 4, body). Source-trajectory dominance ranges 52–100% across five injectable architectures; the metric-sensitivity caveat (action-vector cosine vs. end-effector-velocity classifier) and per-architecture DTW gap are in App. B and App. G.5.
3. **Single-feature ablation tolerance spans the architectures** (Fig. 7). Zero-effect rates span 28% (SmolVLA) to 92% (OFT) over $n = 15{,}096$ concept-task pairs; concepts are encoded across multiple features, not localized in one. Cross-model normalized SAE reconstruction MSE is in Fig. 12.
4. **SAE pooling preference is architecture-tied, not universal** (Tab. 8). Per-token wins on $\pi_{0.5}$, SmolVLA (within noise of mean-pool), and OFT; mean-pool wins on X-VLA and GR00T-DiT. Two regimes emerge across five architectures: *per-token-required* ($\pi_{0.5}$ 81/3, OFT 80/7) where mean-pool collapses below 10%, and *pooling-tolerant* (X-VLA 51/81, SmolVLA 67/66, GR00T-DiT 80/83) where all arms remain functional. The temporal-contrastive arm degrades every architecture and converges to $\cos_{\text{adj}} \to 0.997$ on every architecture with training data (Tab. 8).
5. **Linear probes recover task-relevant information on every probed architecture** (Tab. 15). What gets decoded differs by pathway: expert/action pathways encode state dynamics; VLM pathways encode goal semantics.

Synthesis-level limitations. Tab. 8 now covers all five language-capable architectures. The OFT per-suite breakdown is in App. G.3. DTW corroboration of cross-task source dominance is currently $\pi_{0.5}$-only. The body's pathway-specialization claim (§4.4) is supported by three different architecture-specific metrics rather than one cross-model metric, documented in App. B. The remaining appendix sections fill in each per-architecture detail behind these five claims.

F 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, and concept ablation produces architecture-dependent kill-switches. 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 $\times$ 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).

F.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.

Subspace separability 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, evidence for separability of goal identity from motor execution within the same layer rather than for direct causal control over goal selection. Demonstrating causal goal control would require additive steering along the goal-discriminant directions (rather than subspace replacement) to redirect behavior toward the source goal; we leave that experiment to future work.

F.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.

F.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's output is near-orthogonal to 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.

F.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.

G Additional Experimental Results

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

Fig. 5 reports the 18-layer × 2-pathway × 2-pooling-mode rollout validation on $\pi_{0.5}$/libero_object (n=20 episodes per (layer, pooling) cell; 1,600 episodes in total). The two SAE families are trained on success-only baseline-rollout activations under identical hyperparameters (k=64, 8× expansion, 30 epochs), differing only in whether the latent code is computed per token or after mean-pooling and broadcast back to every position.

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 (99.6% src)	Y (96.4% src)	Y (99.8% src)	Y (52.1% src)	Y (57.0% src)	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
Pref. SAE pooling	per-token	per-token	mean-pool[†]	tied / indifferent[†]	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, `libero_object` immune. *MetaWorld difficulty-dependent. [†]Pooling splits by architecture (§4.6). [‡]GR00T VL-SA layers benefit from mean-pooling.

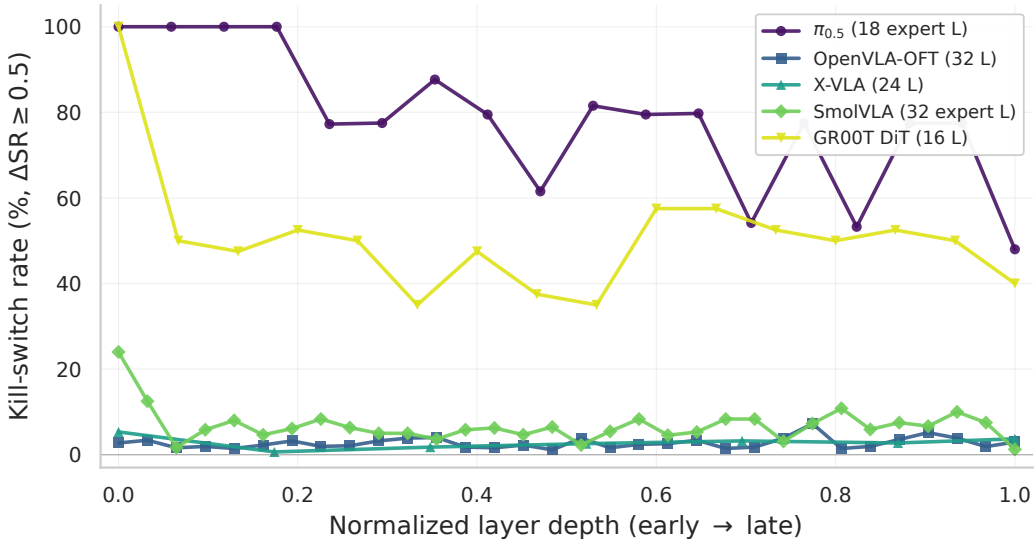


Figure 4: Cross-architecture per-layer kill-switch profile. x-axis: normalized layer depth; y-axis: fraction of (concept, task) ablation pairs with $\Delta\text{SR} \geq 0.5$. $\text{Pi}_{0.5}$ expert and GR00T DiT (action pathways) are 40–100% destructive; OFT, X-VLA, and SmolVLA VLM backbones stay under 10%. Per-architecture pair counts in App. G.

Mean-pool collapse is universal on the expert pathway: success drops from 93.8% baseline to 8.9% averaged across all 18 layers, replicating the Fig. 1 headline. Per-token preservation holds at every expert layer (range 55–100%, 18-layer mean 83.6%). The PaliGemma pathway shows a late-stack reversal: mean-pooled SAEs at L14–L17 preserve 75–85% success while mid-stack layers (L0–L13) collapse to 0%, consistent with PaliGemma's role as goal/scene encoder whose late-layer activations are position-invariant and therefore tolerate pooling. A LIBERO-10 task-by-task comparison on $\pi_{0.5}$ expert (n=250 per task) reaches the same conclusion in stronger form: 1,748/2,500 success under per-token SAE (69.9%) vs. 9/2,500 under mean-pool (0.4%).

G.2 Temporal-Contrastive SAE Pooling (T-SAE Ablation)

A natural reviewer question is whether a temporal-aware SAE that preserves per-token operation but encourages adjacent codes to be similar would close the per-token versus mean-pool gap. We port the Temporal SAE (T-SAE) recipe of Bhalla et al. [2025] to VLA action tokens and evaluate it as a third pooling arm matched on activation source, TopK ($k = 64$), expansion ($8\times$), optimizer, and epoch count against the per-token and mean-pool baselines reported in Fig. 5. The contrastive term is symmetric InfoNCE on adjacent action-token codes ($\alpha = 1.0$, $\tau = 1.0$) added to per-token MSE; details and code in Appendix L.

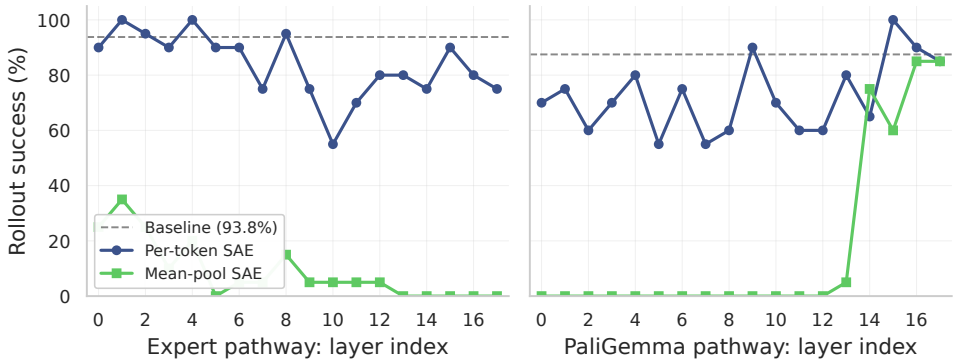


Figure 5: Per-layer rollout success for per-token vs. mean-pool SAE on $\pi_{0.5}$/`libero_object` (n=20/cell). Per-token preserves expert success at every layer; mean-pool collapses at expert L0–L17 and PaliGemma L0–L13, with a reversal at PaliGemma L14–L17. Dashed lines: per-pathway baselines (expert 93.8%, PaliGemma 87.5%).

Cross-architecture training and rollout coverage. We train T-SAE checkpoints on five architectures ($\pi_{0.5}$ expert 18 layers, X-VLA 24 layers, SmolVLA expert 18 layers, GR00T-DiT 16 layers, OFT 32 layers) and validate via rollout on all five.

Pooling preference is architecture-tied. The five-architecture three-arm rollout comparison reveals two pooling regimes rather than a single VLA-wide preference. *Per-token-required regime:* On $\pi_{0.5}$ (Tab. 7, n=1,440 episodes per arm) the dose-response is per-token 81.3% > temporal 71.5% > mean-pool 3.3% (baseline 89.9%): mean-pool collapses 87pp. OFT replicates the same regime over its 32-layer Llama-2 stack: per-token/temporal/mean-pool = $80.1/46.2/7.3\%$ (App. G.3); mean-pool collapses 73pp below per-token. Per-token success is approximately depth-invariant on each LIBERO suite (Fig. 6); the per-suite `libero_10` grand-mean (74.5%) sits 12pp below `object` (86.6%) due to a uniformly lower ceiling on the long-horizon suite, not a layer-localized regression. *Pooling-tolerant regime:* On X-VLA, the ordering inverts to mean-pool > per-token > temporal: per-token/temporal/mean-pool = $50.6/23.5/80.7\%$ over the full 24-layer stack (n=4,800 episodes per arm at 5 ep/task). SmolVLA shows no architecturally-decisive preference: per-token/temporal/mean-pool = $67.2/35.1/66.4\%$ (per-token vs. mean-pool Δ=0.8pp grand-mean, sign flips on `libero_10` and `spatial`). GR00T-DiT also sits in the pooling-tolerant regime: per-token/temporal/mean-pool = $80.0/76.2/82.6\%$ across the 16-layer DiT pathway, with mean-pool exceeding per-token by 2.6pp. The temporal-contrastive arm degrades every architecture in both regimes.

Table 7: $\pi_{0.5}$ expert 3-arm SAE pooling (n=1,440/arm; 18 layers × 4 suites × 20 ep). Deltas vs. Tab. 3 baseline. Per-token > temporal > mean-pool on every suite.

Suite	Baseline	Per-token	Temporal	Mean-pool
libero_goal	93.2%	83.1% (–10.1)	76.1% (–17.1)	8.1% (–85.1)
libero_spatial	92.8%	**90.0%** (–2.8)	**88.3%** (–4.5)	1.9% (–90.9)
libero_object	87.6%	81.9% (–5.7)	74.7% (–12.9)	2.2% (–85.4)
libero_10	86.0%	70.3% (–15.7)	46.7% (–39.3)	1.1% (–84.9)
Mean (4 suites)	**89.90%**	**81.32%**	**71.46%**	**3.33%**

Mechanism (cosine geometry of latent codes). A post-hoc diagnostic on layer 8 (1.5M adjacent-pair samples; 5K held-out) measures the cosine geometry of the trained latent codes $z = \text{encode}(\mathbf{h})$. Per-token natural $\cos(z_t, z_{t+1}) = 0.857$ (adjacent activations encode the same scene/state context, so codes already sit nearby in latent space). Mean-pool $\cos_{\text{adj}} = 0.852$, within 0.005 of per-token (pooling itself does not reshape latent geometry; the destruction is at inference, not training). The temporal-contrastive arm pushes $\cos_{\text{adj}}$ to 0.997, the predicted theoretical optimum of the InfoNCE objective with our normalization. The TopK active-set Jaccard is unchanged at ≈ 0.006 across all three arms: the contrastive loss does not learn to align active feature subsets, it scales the few shared active features so the L2-normalized sparse vectors point in nearly the same direction. The

smoothness prior optimizes for adjacent-code identity at the cost of phase distinction: the InfoNCE loss converges, and the resulting collapse deletes the structure that downstream policy MLPs read.

Cross-architecture confirmation. The contrastive arm converges to $\cos_{\text{adj}} \approx 0.997$ on $\pi_{0.5}$, X-VLA, SmolVLA, and GR00T-DiT (SmolVLA 0.940, slightly under-converged) while keeping per-token-grade R^2 (≥ 0.97). On OFT, training is data-limited ($\sim$26K pairs/layer vs. $\geq$ 100K elsewhere) and $\cos_{\text{adj}}$ rises only to 0.717. Per-architecture per-token reconstruction means and InfoNCE values are folded into Tab. 8.

Table 8: Cross-architecture 3-arm SAE rollout fidelity and T-SAE training. Recon and InfoNCE from matched hyperparameters (α=1.0, k=64, 8×). **Bold**: best SAE arm per row. Per-token wins on $\pi_{0.5}$, SmolVLA, and OFT; mean-pool wins on X-VLA and GR00T-DiT.

Model (pathway)	L	Baseline	Per-token	Temporal	Mean-pool	Recon (PT)	Recon (T)	InfoNCE
$\pi_{0.5}$ (expert)	18	89.9%	**81.3%**	71.5%	3.3%	0.018	0.026	5.26
X-VLA (transformer)	24	95.2%	50.6%	23.5%	**80.7%**	0.008	0.015	5.26
SmolVLA (expert)	32	67.3%	**67.2%**	35.1%	66.4%	0.002	0.108	5.35
OFT (Llama-2)	32	95.4%	**80.1%**	46.2%	7.3%	0.072	0.342	5.55
GR00T (DiT)	16	83.9%	80.0%	76.2%	**82.6%**	0.010	0.008	5.28

Cross-architecture 3-arm rollout summary.

Refined claim. The five-architecture rollouts refine the original per-token-vs-mean-pool finding: which pooling preserves rollout fidelity is an architectural property of the model, not a universal property of VLAs. Two regimes emerge. (i) *Per-token-required:* $\pi_{0.5}$ (PaliGemma cross-attention via the residual stream) and OFT (Llama-2 with continuous L1 action head) encode position-distinct phase information in the residual stream; mean-pooling collapses these (3.3%, 7.3%). (ii) *Pooling-tolerant:* X-VLA (soft-prompted Florence-Large), SmolVLA (interleaved VLM-expert), and GR00T-DiT (diffusion-transformer expert) carry approximately position-invariant action-token activations and tolerate or prefer mean-pooling (X-VLA prefers it at 80.7%, SmolVLA is indifferent within 0.8pp, GR00T-DiT prefers mean-pool at 82.6% vs. 80.0% per-token). The architectural divider is whether action tokens carry phase-distinct information at the SAE injection layer; OFT's similarity to $\pi_{0.5}$ despite a different VLM and action head shows the divider is the residual-stream encoding regime, not the specific backbone. The temporal-contrastive arm degrades every architecture in both regimes because it collapses adjacent latent codes regardless of the underlying pooling preference; the cosine collapse ($\cos_{\text{adj}} \to 1$) is the proximate cause, not reconstruction MSE ($R^2 \geq 0.97$ under temporal pressure).

Limitations. α was not swept; we report the published Bhalla et al. [2025] default $\alpha = 1.0$. The conclusion is therefore that at the published hyperparameters T-SAE degrades VLA rollout fidelity on every architecture we tested; whether a tuned $\alpha < 1$ can encourage temporal smoothness without inducing the latent-code collapse documented in the cosine geometry diagnostic above is an open question. We omit the Matryoshka decoder used in the original T-SAE recipe to keep the comparison strictly to the contrastive term.

G.3 OFT Per-Suite 3-Arm Table

Tab. 9 expands the body grand-means for OFT into a per-suite breakdown, parallel to Tab. 7 for $\pi_{0.5}$. OFT shows the per-token-required regime (mean-pool $\leq 9.2\%$ on every suite). The OFT `libero_10` per-token suite mean (74.5%) is uniformly lower than the short-horizon suites across all 32 layers (Fig. 6) rather than reflecting layer-localized degradation.

G.4 OFT Per-Token Layerwise Profile

G.5 Dynamic Time Warping Analysis of Cross-Task Trajectories

The *cosine-dominance* override metric used in §4.3 (Eq. 2) compares per-step action-vector directions, which is a local geometric measure. We add a global trajectory-shape measure: Dynamic Time

Table 9: OFT 3-arm SAE rollout per-suite (32 layers × 25 ep per cell). Per-token wins on every suite by 26–40pp; mean-pool single-digit on every suite.

Suite	Per-token	Temporal	Mean-pool
libero_object	**86.6%**	54.6%	9.2%
libero_spatial	**78.2%**	51.8%	6.9%
libero_goal	**81.0%**	43.8%	8.0%
libero_10	**74.5%**	34.5%	5.1%
Mean (4 suites)	**80.1%**	46.2%	7.3%

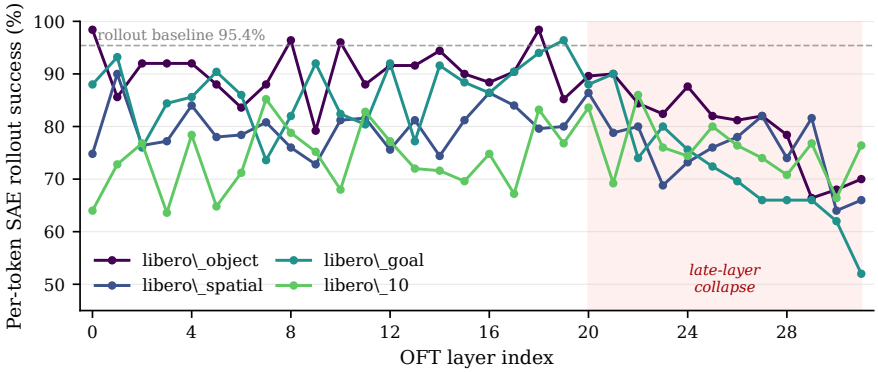


Figure 6: OFT per-token SAE rollout success vs. decoder layer (L0–L31), per LIBERO suite. Per-token success is approximately depth-invariant on every suite. Dashed line: rollout baseline (95.4%).

Warping (DTW) on end-effector position trajectories, normalized by warping-path length. For each cross-task pair (A, B) with prompt B and source A's expert activations injected at layers 16–17, we compute three DTW distances per episode: $d_{\text{src}} = \text{DTW}(\mathbf{x}^{B|A\text{-inj}}, \mathbf{x}^{A})$, $d_{\text{dst}} = \text{DTW}(\mathbf{x}^{B|A\text{-inj}}, \mathbf{x}^{B})$, and the inherent $d_{A,B} = \text{DTW}(\mathbf{x}^{A}, \mathbf{x}^{B})$ as a control. We report the fraction of episodes with $d_{\text{src}} < d_{\text{dst}}$ as DTW source-dominance, paralleling the cosine-dominance override rate.

DTW corroborates the cosine result on `libero_goal` (90.7% source-dominance, n=270, Wilson 95% CI [86.7, 93.6]) and `libero_spatial` (86.4%, n=44). The two metrics agree directionally: the injected trajectory is closer to the source than to the destination on shared-vocabulary suites, both per-step (cosine) and globally (DTW). The 8–12pp gap between cosine and DTW source-dominance on these suites is expected because cosine measures action-vector *direction* step-wise while DTW measures trajectory *shape* globally; an injected episode that briefly aligns with the source then diverges can register as cosine-dominant but DTW-mixed.

The long-horizon `libero_10` suite shows weaker DTW agreement (45.8% source-dominance vs. 95.8% cosine-dominance, n=24): the inherent $d_{A,B}$=0.361 is large and injection-induced trajectory truncation often shortens the rollout, making the truncated trajectory artifactually closer to the shorter destination baseline by raw shape distance. Cosine, normalized per-step, is insensitive to this truncation. Because cosine and DTW measure complementary properties, we report both: agreement on `goal`/`spatial` strengthens the source-dominance claim; the `libero_10` divergence is a documented metric-sensitivity to trajectory length under early termination, not a contradiction of the cosine result.

G.6 OFT Cross-Task Override Per LIBERO Suite

Per-suite OFT displacement. The 77.9% override rate that we report for OFT in the main text is the pooled rate across *all* four LIBERO suites (`goal`, `object`, `spatial`, `10`; 45 pairs × 6 injection-layer conditions per suite, n=1,079 episodes after dropping one truncated trajectory). Table 11 breaks the pooled rate down by suite. Override rates fall in a narrow band of 73.7–81.9%, with overlapping 95% Wilson confidence intervals; `libero_goal` is the highest at 81.9% and `libero_spatial` the lowest at 73.7%. The ∼8 pp suite-to-suite spread is small relative to the per-suite CI width (∼10 pp), so we cannot reject the hypothesis that the underlying override rate is

Table 10: DTW source-dominance for cross-task injection on $\pi_{0.5}$, alongside per-step cosine source-dominance. Brackets are Wilson 95% CIs.

Suite	n	DTW→src	DTW→dst	$\text{DTW}_{A,B}$	DTW src-dom. [95% CI]	Cos src-dom.
LIBERO-Goal	270	0.105±0.055	0.232±0.104	0.190	90.7% [86.7, 93.6]	98.9%
LIBERO-Spatial	44	0.121±0.091	0.160±0.089	0.194	86.4% [73.3, 93.6]	97.7%
LIBERO-10 (long)	24	0.338±0.225	0.225±0.073	0.361	45.8% [27.9, 64.9]	95.8%
Pooled	338	0.124	0.222	–	87.0% [83.0, 90.2]	98.5%

Table 11: OFT cross-task injection per LIBERO suite (n=45 pairs each, single direction). Override rate (Eq. 2): tie-band-conditional $|\text{src}|/(|\text{src}| + |\text{dst}|)$ at τ=0.05. Each pair contributes 6 episodes (one per injection-layer condition: `L0`, `L8`, `L16`, `L24`, `L_last`, `ALL`); a single `libero_object` episode was dropped due to a short trajectory. Tie-inclusive override (denominator over all episodes) is 77.9% pooled (per-suite 73.7–81.9%); the difference is the 20.9% of episodes inside the tie band. "Dest success" is the destination-task success rate under injection, expected to be near zero if injection truly overrides goal-conditioning.

LIBERO suite	n pairs	Episodes	Override rate (95% Wilson CI)	Dest success
`libero_goal`	45	270	100.0% (98.3–100.0%)	0.0%
`libero_object`	45	269	97.0% (93.7–98.6%)	19.7%
`libero_spatial`	45	270	90.8% (86.4–93.9%)	0.4%
`libero_10`	45	270	97.7% (94.7–99.0%)	0.0%
Pooled (all suites)	180	1,079	96.4% (94.8–97.5%)	5.0%

uniform across suites; what we can say is that source-dominant displacement is the majority outcome on every suite, not a `goal`-only artifact.

We treat this as a moderate (rather than strong) universality claim. Each pair is run in a single injection direction (`task_a`→`task_b`), unlike our $\pi_{0.5}$ analysis which sweeps both directions, so within-suite n is roughly half what the $\pi_{0.5}$ pooled rate is built on. The `libero_object` suite has a non-trivial 19.7% destination-task success rate under injection, indicating that on this suite the injected activations sometimes fail to override the prompted task; the override rate there should be read as "among episodes where the policy departed from the destination task at all, displacement still tilts toward the source." Finally, the OFT cosine similarities are computed from end-effector velocity rather than action logits (which were not stored at rollout time), and may underestimate displacement for episodes where action-space steering dominates over Cartesian motion.

G.7 $\pi_{0.5}$ Concept-Feature Analysis

This subsection consolidates three previously separate $\pi_{0.5}$ concept-feature analyses (layer specificity, binary task discrimination, and selection-method comparison) into Tab. 12. Across the seven action/object concepts we tracked, later expert layers (L14–L17) carry the strongest concept-specific signal; each top concept feature classifies its tasks at 100% accuracy on LIBERO-10; and concept-aligned selection (Cohen's $d_f \times \text{freq}_f$) recovers features whose ablation perturbs success much less than action-correlated features selected by output-dimension regression.

Feature selection on $\pi_{0.5}$ libero-object: concept-aligned selection (Cohen's $d_f \times \text{freq}_f$) preserves rollout success under ablation (93% → 93%), while action-correlated selection (linear-probe x,y,z,rpy,gripper) drops -14pp because the chosen feature absorbs per-step motor output statistics rather than concept identity.

Causal Necessity of Linear Probe Directions

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. Cross-architecture single-feature ablation sensitivity is reported in Fig. 7 (n=15,096 concept-task pairs across five models).

Table 12: $\pi_{0.5}$ concept-feature analysis. (a) Layer-wise specificity: later expert layers carry the strongest concept-specific signal. (b) Top concept features classify their tasks at 100% accuracy.

(a) Layer-wise concept specificity (raw $\sum_t \mathrm{ReLU}(z_{f,t,\mathrm{ep}})$ per concept; k = thousands).

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)

(b) Single-feature binary task classification on LIBERO-10.

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%

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% vs. 0–20%), pointing to reliance on manipulation-relevant regions rather than full scene context.

H Ablation Studies and Negative Results

H.1 Temporal Ablation: GR00T Phase-Conditioned Feature Importance

Table 13 shows GR00T N1.5 temporal ablation aggregated over the full 32-layer stack across three LIBERO suites (160 conditions). Early-window ablation (-29.7pp suite-mean) is nearly as severe as full-episode ablation (-30.5pp), while mid- and late-window ablation cause only -13.6 and -13.8pp; the asymmetry holds in every suite. The effect scales with task complexity: `libero_long` (303-step mean) shows -44pp early drop vs. -27pp on `libero_goal` (108-step mean). Layer-type-resolved profiles in our raw logs show DiT layers carry the asymmetry (early-window -50pp, similar to full-episode) while Eagle LM layers are flatter (~ -15pp early); the per-layer-type decomposition is not separately tabulated here. The mechanism is the same as on $\pi_{0.5}$ (App. B Wilcoxon p=0.013): motor programs are committed during trajectory initiation, after which the model tolerates feature ablation.

Table 13: GR00T N1.5 temporal ablation by LIBERO suite (160 conditions across 32 layers).

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%**

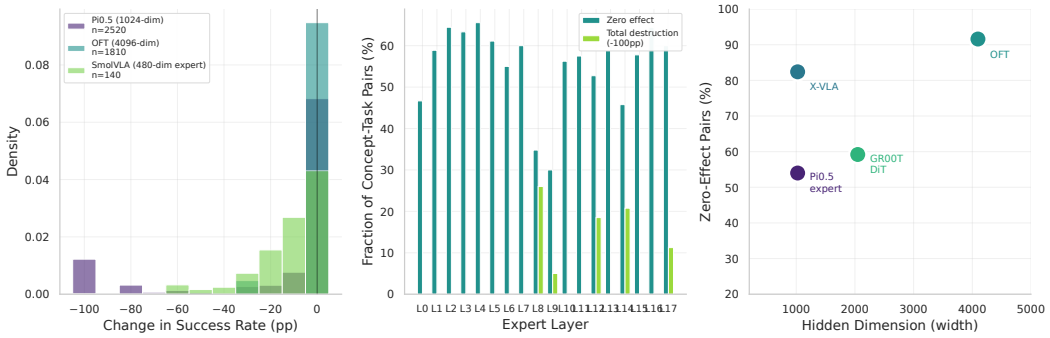
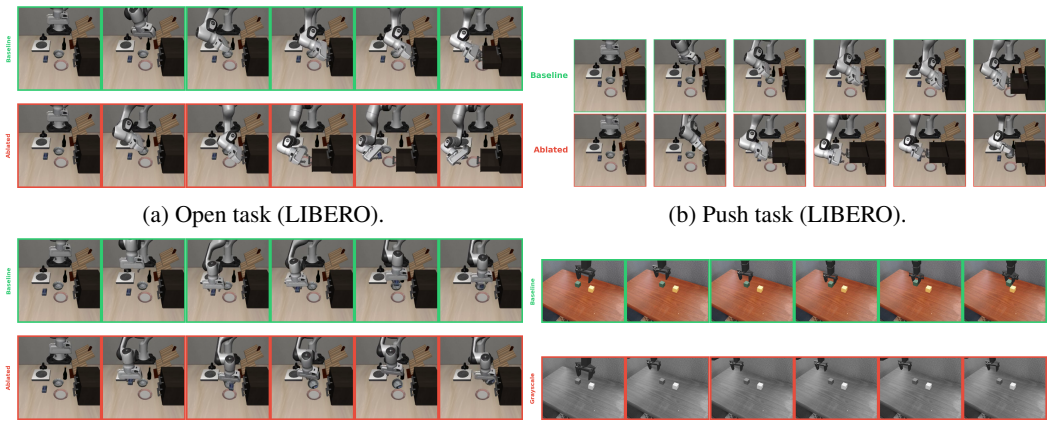


Figure 7: Concept ablation sensitivity across five models. Each bar: fraction of (concept, task) pairs with zero effect (gray), partial (blue), or total destruction (red) under single-feature ablation.

I Qualitative Results and Additional Figures

Across SmolVLA/X-VLA vision perturbations, feature ablation and vision perturbation produce binary failure: the robot either completes the task or fails entirely, with no partial completion observed. Representative examples: SmolVLA 50% center-crop removes button-press spatial context (400-step timeout vs. 67-step baseline); X-VLA grayscale removes color discriminability for the WidowX stack-cube task (0% success). Figure 8 shows representative rollout filmstrips. Frame-by-frame video stills are included in the supplementary material.



(a) Open task (LIBERO).

(b) Push task (LIBERO).

(c) Put task (LIBERO).

(d) Grayscale perturbation (SimplerEnv WidowX).

Figure 8: Baseline (green) vs. ablated (red) rollout filmstrips. (a)–(c): feature ablation on LIBERO collapses task execution to a partial reach. (d): grayscale input on SimplerEnv stack-cube removes color discriminability and the policy fails to grasp.

J Per-Architecture Deep Dives

The bind/select dissociation in §4.5 predicts a model-agnostic diagnostic: expert-pathway disruption produces active misdirection, VLM-pathway disruption produces passive stalling. The architectures we study also show signatures specific to their fusion design and parameter count. Each subsection below opens with the architecture's distinctive failure mode (the bind/select prediction realized in that model's design), then reports per-suite tables and any model-specific figures. Baselines are 25 episodes per task across 4 LIBERO suites (1000 rollouts per model unless noted); they differ by 5–13pp from the smaller-n baselines reported in Tab. 3 for SmolVLA and X-VLA because that table's baselines are per-injection control conditions for cross-condition comparison within the same experiment, while the deep-dive numbers establish a single sample-size-matched baseline across architectures. The ACT control (language-free) is documented in App. C.2; the cross-architecture

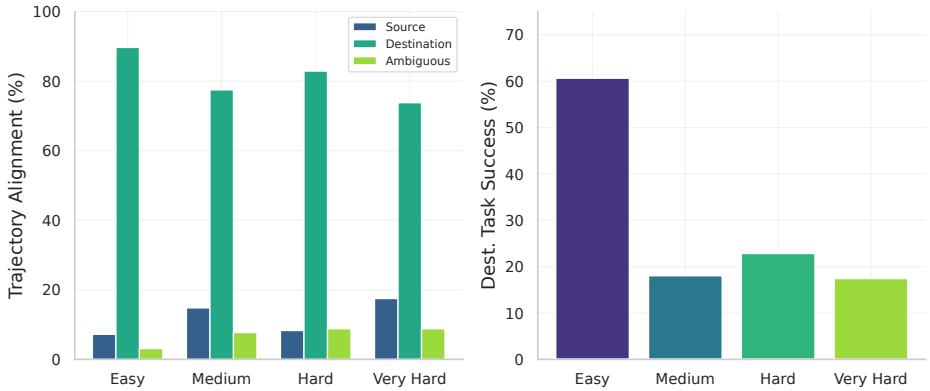


Figure 9: SmolVLA MetaWorld cross-task displacement and injection success by difficulty. Left: source-task override rate (89.7% easy $\rightarrow$ 73.8% very hard). Right: destination success under injection (60.7% easy $\rightarrow$ 17–22% hard).

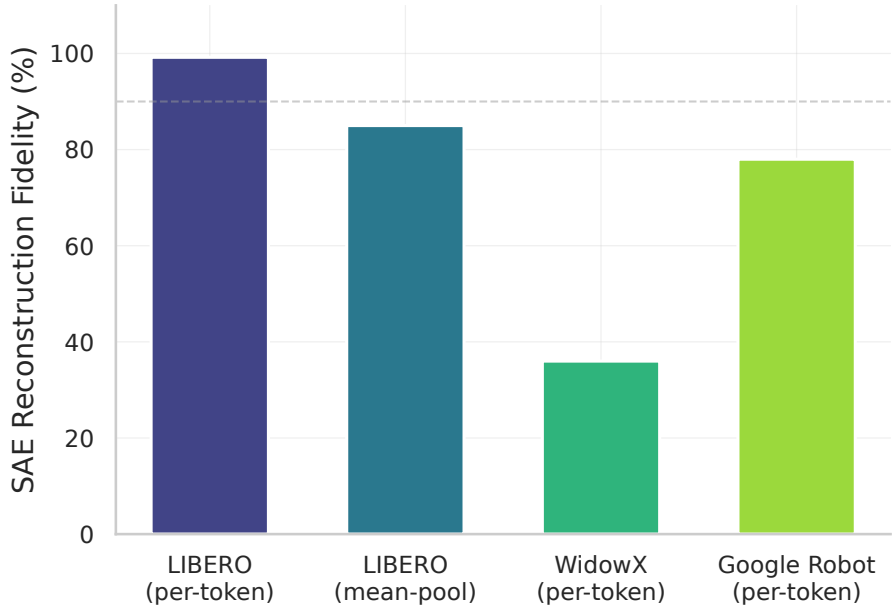


Figure 10: X-VLA SAE reconstruction fidelity across environments. Per-token MSE: 0.008 LIBERO (R^2=0.99) vs. 0.64 WidowX (R^2=0.36); mean-pool raises LIBERO MSE to 0.15.

synthesis is in App. E; the SAE pooling deep dive (per-token vs. mean-pool vs. T-SAE, DTW corroboration, $\pi_{0.5}$ concept-feature analysis) is in App. G.

J.1 OpenVLA-OFT Deep Dive

Failure mode. OFT is the most robust on the baseline at 95.4% overall (libero_object 99.2%, goal 98.8%, spatial 92.4%, libero_10 91.2%). On the same `libero_10` task 8 cluster that produces 0/25 for $\pi_{0.5}$, OFT produces 23/25 (92%), so the brittleness pattern is model-specific rather than scene-specific within this seed range. The dominant OFT failure is structural: zeroing any single residual-stream layer (tested layers 0, 8, 16, 24, 31) drives all four suites to 0%, and recovery via baseline visual-pathway injection plateaus at 14–15% (Tab. 14). The 7B Llama backbone with a continuous-regression head therefore depends on a thin slice of activations through which all task-relevant information must pass, making it the most fragile under per-layer null injection despite being the most robust under raw rollout.

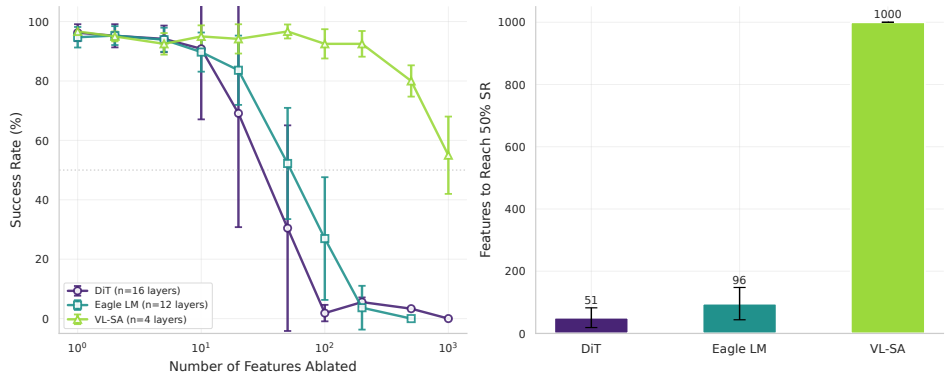


Figure 11: GR00T N1.5 layer-type contribution profiles. DiT layers (16): 40–80% drop; Eagle LM (12): moderate; VL-SA (4): most resilient.

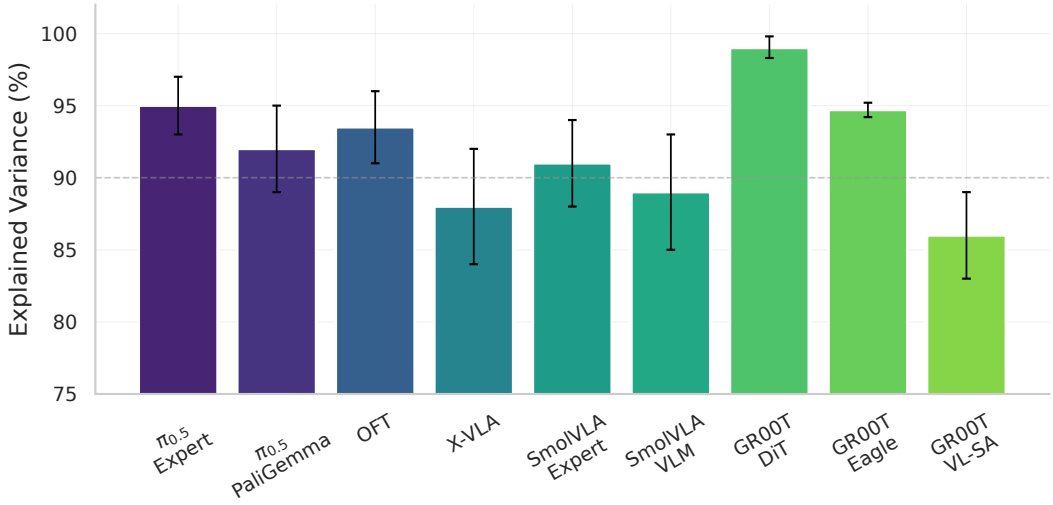


Figure 12: SAE normalized MSE by architecture and pooling. Error bars: layer range within each model. Dashed line: R^2=0.90 threshold.

Per-suite OFT deep dive. Tab. 14 consolidates the four OFT-specific per-suite breakdowns. Zeroing any single layer (tested: 0, 8, 16, 24, 31) under a null prompt destroys success across all suites; injecting zero-baseline visual-pathway activations recovers 14–15% on the three short-horizon suites. Same-scene injection (n=1,518 total episodes) recovers 72–88% per suite. Linear probes for episode length on OFT's 32 layers achieve high R^2 on every suite (0.61–0.99); the spread is small relative to the cross-suite gap and late layers carry slightly higher R^2 on every suite (region aggregates: LIBERO-Spatial early/mid/late 0.984/0.991/0.986; Object 0.719/0.834/0.863; Goal 0.859/0.896/0.933; 10 0.733/0.779/0.827). Temporal-window null injection on `libero_10` (relative to a 9/10-task baseline): early and mid windows destroy all 9 surviving tasks, late nulling rescues 2/9 (Tasks 3 and 9).

J.2 Cross-Model Linear Probing Summary

Table 15 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.

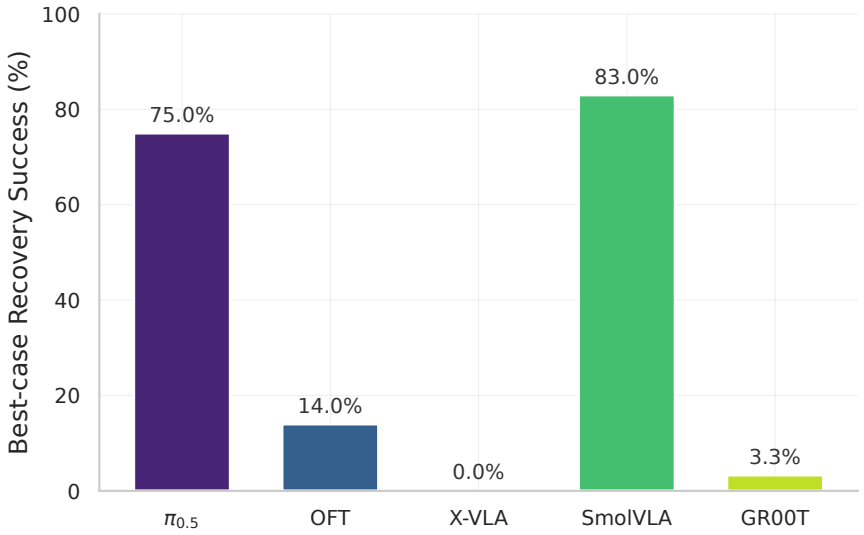


Figure 13: Null-prompt recovery success across architectures: best-case task success when the visual pathway carries no language signal. Conditions per-architecture in App. G.

Table 14: OFT per-suite deep dive. Zero-layer: any of layers 0/8/16/24/31 nulled under empty prompt (baseline ∼90%). Recovery: null + inject zero-baseline activations. Probe R^2: 32 layers, n=149/suite.

	Null injection		Same-scene	Probe R^2 (32 L)	
Suite	**Zero-layer**	**Recovery**	**inj. SR**	**Min**	**Mean**
libero_goal	∼0%	33/234 (14%)	87.8% (n=582)	0.845	0.896
libero_object	∼0%	43/294 (15%)	77.8% (n=162)	0.608	0.813
libero_spatial	∼0%	31/216 (14%)	74.0% (n=192)	0.969	0.988
libero_10	0/10 tasks	–	71.8% (n=582)	0.692	0.780

J.3 $\pi_{0.5}$ Deep Dive

Failure mode. $\pi_{0.5}$ reaches 89.9% overall on the 1000-rollout baseline (libero_object 87.6%, goal 93.2%, spatial 92.8%, libero_10 86.0%). The dominant failure mode is coordinate-cluster brittleness: on `libero_10` task 8 ("put both moka pots on the stove"), one contiguous 25-seed range produces 0/25 success while neighboring 25-seed ranges produce 23/25 (92%) and 20/25 (80%) on the identical task and policy. The 0/25 cluster is not a single bad seed but a contiguous block of initial states that all fail to recover. This brittleness is the surface manifestation of the coordinate-bound motor program identified in §4.3: cross-task injection redirects 99.6% of $\pi_{0.5}$ episodes onto source-task trajectories regardless of the destination scene, and the LIBERO-PRO position perturbations of Zhou et al. [2025] drop $\pi_{0.5}$ from 0.92–0.98 to 0.08–0.38 across all four suites. The same mechanism produces both effects.

Counterfactual prompting by suite. Per-suite ANOVA on $\pi_{0.5}$ counterfactual rollouts confirms the aggregate null result: `libero_object` (n=1,496, 22 prompt categories) $p > 0.24$; `libero_spatial` (n=353, 9 categories) $p > 0.067$; `libero_goal` (n=380, 6 categories) $p > 0.24$. `libero_10` (n=1,167) sits inside the aggregate p=0.247 envelope of Tab. 5.

Cross-task injection per-suite. Pathway-specific cross-task conditions on $\pi_{0.5}$ all collapse destination success: GOAL baseline 91.5% → 0.0% under cross-prompt without injection (n=236/condition; PaliGemma-ALL 0.4%, Expert L16 1.3%); SPATIAL (n=34) drops 100% → 64.7% under cross-prompt and 20.6% under own-prompt + PaliGemma-ALL injection; LIBERO-10 (n=24) drops 62.5% → 0% across all injection conditions.

Table 15: Cross-model linear probing summary. Oracle ratio: probe R^2 / oracle R^2 (fraction of ground-truth state linearly decodable from activations). Pathway-pair quantification (expert/VLM oracle ratio) is on disk for SmolVLA only; $\pi_{0.5}$ and GR00T pathway split is established qualitatively in §4.5.

Model	Probe Target	Metric	Result
$\pi_{0.5}$ (expert)	State prediction	R^2	0.45
$\pi_{0.5}$ (PaliGemma)	Prompt classification	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%

J.4 X-VLA Deep Dive

Failure mode. X-VLA reaches 95.2% overall on the 1000-rollout baseline (libero_object 98.8%, goal 98.8%, spatial 95.6%, libero_10 87.6%) and matches the cross-task injection signature of $\pi_{0.5}$ (99.8% on LIBERO; §4.3) under the same coordinate-bound motor program. X-VLA also exhibits the seed-cluster brittleness pattern observed on $\pi_{0.5}$: on `libero_10` task 8, the original seed range produces 0/25 while a neighboring 25-seed range recovers to 24/25 (96%) on the identical task. Its main failure signatures are visual rather than coordinate-cluster: vision perturbations produce binary outcomes (full success or 0%, no partial completion), and grayscale conversion drops X-VLA's WidowX stack-cube task to 0%, indicating reliance on color-discriminable affordances rather than shape priors. SAE evaluation prefers mean-pool over per-token (mean-pool 81% vs. per-token 51% across the full 24-layer stack), the opposite of $\pi_{0.5}$; this matters operationally because per-token interpretability tooling that ports cleanly to $\pi_{0.5}$ damages X-VLA reconstruction quality, and conversely.

Per-suite results. X-VLA per-suite baselines (Tab. 3, n=250/suite): goal 98.8%, object 98.8%, spatial 95.6%, LIBERO-10 87.6%. Table 16 summarizes grid ablation, counterfactual prompting, and concept ablation by suite.

Table 16: X-VLA per-suite breakdown. Zero any layer: any of 24 tested layers nulled. Null prompt: success under empty string. Zero eff./Δpp: concept ablation zero-effect rate and mean delta. n=4,800 counterfactual; n=2,480 concept-ablation pairs.

Suite	Baseline	Zero any layer	Null prompt	Zero eff.	Δpp
Goal	98.8%	0%	10%	82.6%	−2.2
Object	98.8%	0%	60%	98.2%	−1.3
Spatial	95.6%	0%	48%	85.2%	−2.6
LIBERO-10	87.6%	0%	28%	74.7%	−2.4

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).

J.5 SmolVLA Deep Dive

Failure mode. SmolVLA reaches 67.3% overall on the 1000-rollout baseline (libero_object 87.6%, goal 74.8%, spatial 65.6%, libero_10 41.2%) and shows two distinct failure modes. (i) Multi-step task collapse: `libero_10` task 0 (alphabet soup + tomato sauce, 2-step) and task 4 (white mug + yellow mug, 2-step) sit far below the rest of the suite, while single-step tasks on the same suite stay above 60%. (ii) Genuine model weakness on the same coordinate cluster that produces 0/25

for $\pi_{0.5}$: SmolVLA also gets 0/25 there, and unlike $\pi_{0.5}$ recovers only to 12% and 60% on the two neighboring seed ranges (vs. $\pi_{0.5}$'s 92% and 80%). The architectural caveat from §4.3 also lives here: the interleaved VLM-expert fusion attenuates cross-task override to 52.1% on LIBERO (vs. 99.6–99.8% for $\pi_{0.5}$/X-VLA), with the expert pathway carrying the residual signal at 78.6% override and the VLM pathway at 25.6%. Vision perturbations also bite: a 50% center crop on a button-press task drives SmolVLA from a 67-step baseline rollout to a 400-step timeout.

Per-suite results. SmolVLA per-suite baselines (Tab. 3, n=250/suite): goal 74.8%, object 87.6%, spatial 65.6%, LIBERO-10 41.2%. Grid ablation across 65 conditions (32 expert + 32 VLM + baseline) shows that expert-layer zeroing maintains partial success (Table 17), while concept ablation reveals stronger sensitivity than OFT or X-VLA.

Table 17: SmolVLA per-suite breakdown. Expert zero: success-rate range when zeroing individual expert layers. Zero eff./Δpp: expert-pathway concept ablation (n=1,696). VLM concept ablation (n=210): goal -10.7pp, object $+6.4$pp, spatial -8.5pp, LIBERO-10 $+5.9$pp.

Suite	Baseline	Expert zero	Zero eff. (exp)	Δpp (exp)
Goal	74.8%	60–83%	14.6%	-4.7
Object	87.6%	60–83%	0.0%	-3.9
Spatial	65.6%	47–77%	32.7%	-17.7
LIBERO-10	41.2%	0–33%	20.3%	$+0.8$

SmolVLA expert concept ablation is most destructive on `libero_spatial` (-17.7pp, 32.7% zero effect) and least on `libero_10` ($+0.8$pp, 20.3% zero effect). VLM concept ablation shows a different pattern: `libero_goal` is most affected (-10.7pp) while `libero_object` shows positive delta ($+6.4$pp), consistent with the VLM encoding goal semantics.

J.6 GR00T N1.5 Deep Dive

Failure mode. GR00T reaches 83.9% overall on the 1000-rollout baseline (libero_object 95.6%, goal 93.6%, spatial 69.2%*, libero_10 77.2%) and shows a layer-specific failure signature within its multi-pathway design. Zeroing any of the 16 DiT layers drives all four suites to 0%, while VL-SA layers tolerate zeroing with 40–80% residual success (Fig. 11), supporting the bind/select prediction that motor-expert pathways carry the primary load. Suite-level fragility tracks task complexity: under non-baseline prompt categories, `libero_goal` collapses from 96.7% to 18.9% while `libero_object` retains 73.3%, and concept ablation finds `libero_long` most destructible (42.2% zero-effect rate, 11.0% destruction). The DiT-as-expert / VL-SA-as-VLM split makes GR00T the model where pathway-specific failure attribution is most actionable for diagnosis. The original `liorbenhorin-nv/groot-libero_goal-64_40000` checkpoint was deleted from HuggingFace before our final batch, so the goal results use `Tacoin/GR00T-N1.5-3B-LIBERO-GOAL` (GR00T native format) and the libero_10 results use `liorbenhorin-nv/groot-libero_10-64_40000` (LeRobot format), both substituted after a brief trial of `aractingi` replacements yielded $\leq$10% baselines we attribute to weaker public exports. The 69.2%* on `libero_spatial` uses a public N1.5 spatial export at batch size 128 / 20K steps; the NVIDIA Isaac-GR00T LIBERO README cites 97.65% (195/200) on libero_spatial under a much larger batch size 640 / 20K steps configuration, and community fine-tune reports span 68–94% depending on modality settings, so our 69.2% sits at the low end of that range. Because this falls well short of the published reference, we ran limited experiments on `libero_spatial` for GR00T and report the baseline alone in Tab. 3 (†); the qualitative findings (DiT load-bearing, VL-SA tolerant) remain consistent across the suites with full coverage.

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 18 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

Table 18: GR00T N1.5 per-suite breakdown. Zero DiT: any of 16 DiT layers nulled. CF "other": counterfactual success under non-baseline prompts (n=180/suite). Zero eff./Δpp: concept ablation across 6,500 pairs.

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

(42.2%) and highest destruction rate (11.0%), reflecting the greater fragility of complex multi-step tasks.

J.7 Operational Summary: Failure-Attribution Rules

For runtime diagnosis, three failure-attribution rules generalize across the architectures studied. (1) If a rollout reaches the wrong target location with confident motion, suspect expert-pathway corruption (consistent with the coordinate-binding evidence in §4.3). (2) If a rollout stalls without committing to any motion, suspect VLM-pathway corruption (consistent with the goal-encoding evidence in App. F.1). (3) If success rate collapses on a contiguous seed range while neighboring ranges succeed, suspect coordinate-cluster brittleness in the absolute-position substrate, not a model-wide regression. The first two rules apply to multi-pathway architectures ($\pi_{0.5}$, SmolVLA, GR00T); rule (3) applies to any model whose action distribution is dominated by absolute coordinates (here $\pi_{0.5}$ and SmolVLA, with OFT resilient on the same seed range).

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.

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.

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 $4\times$ or $8\times$ expansion. Sparsity is enforced via TopK with $k = 64$ active features per token, and decoder weights are tied to the encoder transpose ($\mathbf{W}_d = \mathbf{W}_e^\top$).

K.2 Benchmarks

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.

K.3 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 420,000+ rollout episodes with videos.

K.4 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 Temporal-Contrastive SAE Training Details

We follow the T-SAE recipe of Bhalla et al. [2025], modified only in the choice of TopK ($k = 64$) and expansion factor ($8\times$) to match our existing per-token and mean-pool baselines. The total loss adds a symmetric InfoNCE contrastive term to the per-token reconstruction MSE on adjacent latent codes:

$$\mathcal{L} \;=\; \mathcal{L}_{\text{recon}} \;+\; \alpha\,\mathcal{L}_{\text{contr}}, \qquad \mathcal{L}_{\text{contr}} \;=\; -\tfrac{1}{N}\sum_i \log \frac{\exp s(z_t^i, z_{t-1}^i)/\tau}{\sum_j \exp s(z_t^i, z_{t-1}^j)/\tau} \;+\; \text{(symmetric)}, \tag{11}$$

with $s(\cdot,\cdot)$ cosine similarity, $\alpha = 1.0$, $\tau = 1.0$. VLA activations are 3-D (T timesteps $\times$ S tokens-per-step $\times$ D); we use `cross_time` adjacency $(t, s) \leftrightarrow (t+1, s)$ as the most direct port of T-SAE's adjacent-token InfoNCE.

Matched ablation. Per-token, mean-pool, and temporal arms differ only in pooling and the contrastive term: same activations, same successes-only filter, same TopK k=64, same $8\times$ expansion, same Adam (lr $= 10^{-3}$, batch 512, 100 epochs), same per-position normalization, and same checkpoint format. The $\pi_{0.5}$ 3-arm rollout validator (Tab. 7) is a single launcher that swaps SAE directories.

Risks and confounds. TopK k=64 deviates from the T-SAE paper's BatchTopK k=20 to match our v3 baselines; any TopK-induced effect cancels in the temporal-vs-per-token contrast. In-batch negatives reuse other batch elements as negatives even from the same episode at a different timestep, identical to the original T-SAE setup. We omit the Matryoshka coarse-to-fine decoder used in the original recipe to isolate the contrastive term.

M Broader Impact

This work is interpretability research aimed at making VLA decision-making auditable. The artifacts we release (520+ SAE checkpoints, 79 identified concepts, the cross-task-injection rollout corpus, and the feature-exploration platform) are debugging tools rather than deployment-ready policies, and all experiments run in simulation.

Positive impact. Mechanistic decomposition supports principled failure analysis: kill-switch mapping localizes the features and layers a VLA depends on, and spatial-binding analysis flags coordinate-grounded brittleness before deployment. Robotics operators currently have no principled tool for diagnosing why a VLA-controlled robot deviates from intended behavior; the interventional protocols here (null-injection, cross-task injection, single-feature ablation) narrow a failure to a pathway, a layer, or a feature subspace, which is a prerequisite for safety certification.

Risks and dual use. Activation injection and feature steering redirect a deployed policy's behavior without retraining, so an attacker with white-box access to a VLA's residual stream can reroute a manipulation arm to a different target. We mitigate this exposure by reporting only LIBERO, MetaWorld, SimplerEnv, and ALOHA simulation results, by noting that real-world deployment requires fine-tuning that reshapes the representations we analyze, and by withholding a turn-key real-robot exploit harness. Independently, the coordinate-bound binding documented in §4.3 is itself a deployment hazard: VLAs that pass simulation benchmarks fail under modest scene-position perturbations [Zhou et al., 2025], which is consequential for medical and autonomous-driving-adjacent manipulation. Practitioners deploying VLAs for high-stakes tasks should run perturbation-style robustness benchmarks [Zhou et al., 2025, Fei et al., 2025] before deployment.

Compute and energy. Total compute is reported in §K.3; experiments span $\sim$7.1 TB of activation, rollout, and SAE-checkpoint data. We trained 520+ distinct SAEs and ran 420,000+ rollout episodes. All training used existing pretrained VLA checkpoints; we do not pretrain new VLAs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The four claims listed in the abstract and Section 1 (visual pathway dominance, suite-dependent language sensitivity, coordinate-bound motor programs via cross-task injection, and per-token vs. mean-pool vs. temporal-contrastive SAE pooling) are each supported by named experiments in Sections 4.2–4.6 with sample sizes, p-values, and Wilson 95% CIs reported.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations appendix (Appendix B) discusses benchmark scope (simulation-only), counterfactual prompt coverage, cross-task injection confounds, and steering sensitivity / concept identification caveats. The intervention-hook protocol (Appendix D.4) documents per-architecture hook placement (full `DecoderLayer` output vs. `layer.mlp` output) and clarifies which absolute magnitudes are full-layer-hook artifacts versus MLP-only-confirmed effects.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper is empirical; we present no theorems requiring proof. The SAE objective (Eq. 5), the override metric (Eq. 2), and the temporal-contrastive loss (Eq. 11) are operational definitions accompanied by experimental validation, not theoretical claims.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3, Appendix D, and Appendix L specify SAE architecture (TopK k=64, 4×/8× expansion, tied decoder, AuxK with $\alpha_{\text{aux}} = 1/32$), training (500K samples, batch 4096, 100 epochs, lr 3×10^{-4} cosine), the four injection conditions, the six counterfactual prompt categories, the three intervention-hook placements, and the temporal-contrastive matched-ablation protocol. Per-experiment sample sizes are in figure/table captions.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code, SAE checkpoints, the rollout-validation harness, and the feature-exploration platform (Action Atlas, https://action-atlas.com; identifying content anonymized for review) as supplementary material. Reproduction instructions, dependency specs, and per-experiment launchers are included in the supplementary ZIP.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters and choices are reported in Section 3, Appendix D.1, and Appendix L. Episode counts per experiment type and per architecture are in Table 2. The intervention-hook protocol distinguishing full-layer-residual, residual-additive, and MLP-only placements is documented in Appendix D.4.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All binary success metrics use Wilson 95% confidence intervals (Section 3); ANOVA effect sizes are reported as η^2. Specific statistics include $\pi_{0.5}$ prompt-category ANOVA $F(4, 3391) = 1.23$, $p = 0.247$, $\eta^2 = 0.0015$ (§4.4); cross-task override Wilson 95% CI ± 0.4pp at n=1,968 (Tab. 4); phase-specific $\pi_{0.5}$ transport-phase steering Wilcoxon $p = 0.013$ (App. B); random-direction control override Wilson 95% CI [36.6, 63.4] at n=50, indistinguishable from chance (App. C.3).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute budget is reported in Appendix K.3: 8×A100-80GB for $\pi_{0.5}$/OFT/large-scale SAE/cross-model work, RTX 5090 (32GB) for GR00T, 2×RTX 4090 (24GB) for SmolVLA/X-VLA. Total artifacts ∼7.1 TB (4.3 TB activations, 152 GB SAE checkpoints, 420,000+ rollouts).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

Answer: [Yes]

Justification: The work uses publicly available simulation benchmarks (LIBERO, MetaWorld, SimplerEnv, ALOHA) and publicly released VLA model weights. No human subjects, no personal data, no scraped private content. Dual-use considerations are addressed in the Broader Impact section.

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Broader Impact section (Appendix M) discusses positive impact (failure-mode debugging tools for VLA operators, brittleness diagnostics), risks (dual-use of activation steering for unintended behavioral redirection), and recommends robustness benchmarking (e.g., LIBERO-PRO) prior to high-stakes deployment.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

Answer: [Yes]

Justification: We release simulation-only artifacts (SAE checkpoints over LIBERO/MetaWorld/SimplerEnv/ALOHA activations and rollout videos) and do not release a turn-key real-robot exploit harness. The released SAEs operate on activations of pretrained VLAs; users must obtain VLA weights separately under the original licenses.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and respected?

Answer: [Yes]

Justification: All six VLA model architectures ($\pi_{0.5}$, OpenVLA-OFT, X-VLA, SmolVLA, GR00T N1.5, ACT) and all four benchmarks (LIBERO, MetaWorld, SimplerEnv, ALOHA) are cited with their original publications. We use checkpoints under their respective public licenses (Apache-2.0 / MIT for the simulation benchmarks; model-specific licenses per the original VLA releases).

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release 520+ SAE checkpoints (per-token, mean-pool, and temporal-contrastive variants across six VLAs), the rollout-validation harness for 3-arm pooling comparison, the feature-exploration platform (Action Atlas), and the cross-task injection rollout corpus. Each is documented in the supplementary materials with directory layouts, filename conventions, and reproduction instructions.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work involves no human subjects or crowdsourcing.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work involves no human subjects.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

Answer: [Yes]

Justification: LLM-based auto-interpretability is one tool offered in the feature-exploration platform for narrating SAE features; it is not part of the core experimental claims of the paper. The core methods (activation injection, SAE training, linear probes, counterfactual prompting) do not depend on LLMs.