SafeSAE-VLA: Interpreting OpenVLA Progress Dynamics with Sparse Feature Analysis

Socrates Osorio[1]* and Joy Zheyun Yang[2]

[1] Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA
socratesj.osorio@berkeley.edu
[2] University of Oxford, Oxford, United Kingdom
joy@robots.ox.ac.uk

Abstract. Understanding whether modern vision-language-action policies internally represent task progress is central to interpretable robot control. This paper presents a progress-based SafeSAE-VLA analysis of OpenVLA on 750 LIBERO episodes, replacing a degenerate binary target with a suite-normalized high-vs.-low split over *relative geometric progress* and testing layer-20 sparse-autoencoder (SAE) features ($d_{\text{sae}} = 16{,}384$) with non-parametric differential analysis and false-discovery-rate (FDR) control. The recovered signal is strong and sparse. Of 1,881 active features, 1,117 are significant (FDR < 0.05), a linear probe reaches 0.918 AUROC (area under the ROC curve), and top-20 features alone retain 0.894 AUROC. Performance is stable across suites (up to 0.985 on `goal` and 0.947 on `object`), while motion-only controls are markedly weaker (0.572–0.711), indicating separability is not explained by gross movement magnitude. Dense raw-activation probes are equal or stronger than SAE readouts (e.g., raw LR 0.975 vs. full-SAE 0.947 under a pooled analysis with bootstrap confidence intervals). We therefore position the contribution not as predictive dominance but as showing that nearly all of the progress signal survives in a compact set of *named, inspectable, intervention-compatible* directions. Layer sweeps, split-robustness checks, and a success-labeled audit (in which the progress-tuned top-20 correctly scores an at-chance 0.471 while the broader SAE basis reaches 0.968) support this scoping, and direct on-OpenVLA feature-setting interventions produce specific, task-local behavioral shifts rather than general closed-loop repair.

Keywords: Vision-Language-Action Models · Sparse Features · Robot Progress Analysis · Mechanistic Interpretability

1 Introduction

Vision-language-action (VLA) models are emerging as general-purpose robot controllers capable of interpreting natural language instructions and generating

* S. Osorio and J. Z. Yang contributed equally to this work.

low-level actions from visual observations [3,14,21]. As these models are deployed in real manipulation loops, a central interpretability question is no longer just *what action is produced*, but *what internal structure separates successful progress from stalled behavior*.

Most prior monitoring formulations in this setting rely on coarse labels or hand-crafted sensor thresholds [9,20]. In our rollout set, a binary violation framing is statistically weak because violation incidence is near-universal, leaving little discriminative variance. This motivates a representation-first re-framing: analyze internal activations against *relative task progress* rather than a collapsed binary label.

We show that policy progress in OpenVLA is encoded in a sparse, linearly decodable subspace of residual representations, recoverable via SAE features and predictive of task progress with AUROC 0.918.

To our knowledge, this is the first study showing that VLA policy progress is encoded in sparse SAE feature directions and can be linearly decoded without policy retraining.

Operationally, we ask: *Do OpenVLA internal activations linearly separate high-progress from low-progress manipulation episodes, and which ranked sparse directions carry this signal most strongly?*

From a representation-analysis standpoint, we treat a modern vision-language-action policy as a visual decision backbone and test whether task progress is recoverable from its internal states.

To investigate this question, we introduce a progress-based SafeSAE-VLA analysis pipeline: we partition episodes into high-progress and low-progress quartiles, encode layer-20 residual states with a trained SAE, run Mann–Whitney differential testing with FDR correction over SAE features, rank dimensions by a composite effect-significance score, and evaluate linear progress readouts on the resulting feature subsets and control baselines.

Our analysis reveals that progress information is *strong*, *linearly decodable*, and *sparsely concentrated*. Across 1,881 active SAE features, 1,117 pass FDR < 0.05, and a logistic probe reaches 0.918 AUROC overall. Critically, the top-20 ranked features retain 0.894 AUROC, showing that most separability mass is concentrated in a small subset of internal directions. We stress that dense raw-activation probes are equal or stronger predictors (Sec. 4.6). The value of the sparse basis is not peak accuracy but that the same signal is carried by a few *named, inspectable, writable* directions.

Contributions.

1. **Discovery: sparse progress code in VLA representations.** We show that policy progress is encoded in a sparse set of SAE features that linearly separate high-progress from low-progress episodes.
2. **Inspectable sparse directions, not predictive dominance.** We identify 1,117 significant progress-associated SAE features (FDR < 0.05), with 0.918 AUROC overall and 0.894 AUROC using only top-20 ranked features. Dense and PCA baselines match or exceed these numbers, so we position the sparse

basis as an interpretability and intervention interface rather than a stronger classifier.
3. **The signal is not motion-magnitude-only.** Motion-only controls (action magnitude / end-effector velocity) underperform SAE-feature probes, while suite-stratified results, layer sweeps, and split-robustness checks support interpretable structure beyond trivial movement statistics.
4. **Direct on-OpenVLA intervention evidence.** Setting top-ranked features to their high-progress class means shifts an independent raw-activation readout far beyond matched-random controls, and a pre-registered specificity batch shows task-local behavioral change without success repair. A success-labeled audit shows the progress-tuned top-20 is a progress probe (0.471 success AUROC), not a semantic-completion detector, so we scope claims to relative geometric progress.

Paper organization. Section 2 reviews mechanistic interpretability and robot-control analysis. Section 3 describes the progress-label construction and differential pipeline. Section 4 presents quantitative and feature-level results. Section 5 discusses interpretation, robustness, and limitations.

2 Related Work

Mechanistic Interpretability. Recent work has demonstrated that neural network internals can be decomposed into interpretable components. Olah et al. [18] introduced the circuits framework for understanding features in vision models. Elhage et al. [11] showed that models represent more features than they have dimensions through superposition, motivating the use of sparse dictionaries to recover monosemantic features. Bricken et al. [4] and Cunningham et al. [8] demonstrated that sparse autoencoders (SAEs) can extract interpretable features from language model activations, with Templeton et al. [19] scaling this approach to production-grade models. Conmy et al. [7] developed automated methods for discovering computational circuits. Our work brings these mechanistic interpretability tools to embodied AI, applying SAEs to VLA residual streams to recover progress-related structure rather than to analyze language understanding.

Robot Safety Monitoring. Safety in robot learning has been addressed through constrained optimization [9], learned recovery policies [20], and comprehensive safe reinforcement learning frameworks [6, 12]. Amodei et al. [1] articulated concrete safety problems including safe exploration and distributional shift, while Hendrycks et al. [13] catalogued open problems in ML safety more broadly. Most of these monitors operate as black-box classifiers over observations or latent states, providing scalar predictions without exposing the internal features driving them. We take this literature as motivation but study a different question. Rather than predicting a binary risk label, we ask whether an interpretable internal quantity, relative task progress, is decodable from decomposed representations, enabling feature-level attribution and inspection.

Vision-Language-Action Models. The VLA paradigm unifies perception, language understanding, and action generation in a single model. RT-1 [5] introduced transformer-based robot control, followed by RT-2 [21] which demonstrated that vision-language pretraining transfers to robotic control. OpenVLA [14] provides an open-source 7B-parameter VLA fine-tuned on the Open X-Embodiment dataset. π_0 [3] introduces flow matching for action generation with a vision-language backbone. Octo [17] and PaLM-E [10] represent additional architectures in this rapidly growing family. Despite their increasing deployment, the internal progress representations of these models remain largely unexplored. SafeSAE-VLA provides a first systematic analysis.

Distinction from Black-Box Failure Detection. Our work complements rather than replaces classifier-based failure detection. A black-box multilayer perceptron (MLP) trained on raw activations achieves equal or higher predictive performance (as we confirm in Sec. 4.6), but provides no insight into which directions encode progress information or how that structure is organized. SAE-based analysis instead exposes a sparse, interpretable feature basis whose individual directions can be inspected, ranked by differential effect, intervened on, and analyzed for stage-dependent dynamics. This interpretability is the point: it reveals *which* internal directions carry progress-related structure and supports direct feature-level interventions, informing future work on interpretable monitoring rather than serving as a deployed classifier.

3 Method

SafeSAE-VLA is a five-stage pipeline for interpretable progress analysis of VLA models: (1) rollout collection with activation caching, (2) progress relabeling, (3) differential feature analysis, (4) runtime monitoring, and (5) feature inspection. Figure 1 provides an overview.

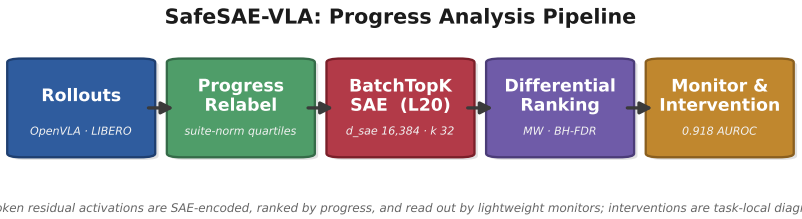

Fig. 1: Progress-based SafeSAE-VLA pipeline. Rollouts are collected from OpenVLA and relabeled into high-progress vs. low-progress quartiles. Residual stream activations at action tokens are differentially analyzed to identify progress-associated directions, which are then used in lightweight progress monitors.

3.1 Progress Label Construction

For each episode, we compute a scalar progress score from available simulator telemetry and normalize it within task suite to control for scale mismatch. We then pool episodes globally and define a clean binary split: top quartile as high-progress ($y = 1$), bottom quartile as low-progress ($y = 0$), and middle quartiles discarded. This extreme-group design yields a high-contrast, variance-preserving target for differential representation analysis.

3.2 Activation Caching

We register forward hooks at target transformer layers in the VLA backbone and cache residual stream activations at action token positions. Let $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^d$ denote the hidden state at timestep t and layer ℓ, extracted at the action token position. For OpenVLA ($d = 4096$), we cache activations at layers $\ell \in \{16, 20, 24\}$ spanning mid-to-late processing stages. Activations are stored in memory-mapped format for efficient SAE training.

3.3 Representation Space

We load the recovered layer-20 SAE checkpoint ($d_{\text{in}} = 4096$, $d_{\text{sae}} = 16384$, $k = 32$) and encode sampled timestep activations into sparse latent features. Differential testing is then performed directly in SAE feature space, preserving the same statistical procedure while restoring feature-level interpretability and compatibility with intervention analyses.

3.4 Differential Activation Analysis

To identify features associated with policy progress, we perform per-feature statistical testing between low-progress and high-progress groups using the Mann–Whitney U test [16].

We apply Benjamini–Hochberg FDR correction [2] across all tested features to control the false discovery rate at $\alpha = 0.05$. For each significant feature, we compute the rank-biserial effect size r and a composite score combining effect size and significance:

$$\text{score}(f) = |r_f| \cdot \left(-\log_{10}(p_f^{\text{adj}} + \epsilon)\right), \tag{1}$$

where $\epsilon = 10^{-300}$ prevents numerical underflow. Features are ranked by this composite score to identify the most progress-relevant directions.

3.5 Runtime Monitor

We evaluate three monitoring approaches using the identified progress features:

- **Raw Activation LR**: L_2-regularized logistic regression (LR) on pooled layer-20 activations.

- **Top-20 Feature LR**: logistic regression using only the top-20 differential features.
- **Random**: uniform random predictions calibrated to the marginal positive-label rate.

All learned monitors are evaluated with stratified splits. We report AUROC, F1, precision, recall, and precision-recall AUC (PR-AUC). We additionally report per-suite progress AUROC.

3.6 Progress-Quartile Analysis

Since the standard safe/unsafe split is degenerate on this dataset, we instead construct a progress-based label with meaningful variance. Episodes are ranked by a suite-normalized progress proxy and partitioned into quartiles: top quartile as high-progress ($y = 1$), bottom quartile as low-progress ($y = 0$), and the middle 50% discarded. This extreme-group design creates a clean contrast for differential testing and linear readout evaluation.

4 Experiments

4.1 Experimental Setup

Model and benchmark. We evaluate on OpenVLA [14] over 750 LIBERO episodes [15] spanning `spatial`, `object`, `goal`, and `long`. Rollouts provide cached layer activations and simulator telemetry.

Progress labels. We replace the invalid safe/unsafe split with a suite-normalized quartile split: top 25% episodes are high-progress, bottom 25% are low-progress, and the middle 50% are dropped. This yields 374 labeled episodes (187/187 balanced).

Differential analysis. We run Mann–Whitney tests on layer-20 SAE features ($d_{\text{sae}} = 16{,}384$) after sparse encoding of sampled timesteps, then apply BH-FDR correction. Features are ranked by the composite effect-significance score in Eq. 1.

Dataset statistics. Table 1 reports per-suite episode counts for labeled classes, total timesteps per class, and progress-score distribution statistics.

4.2 Main Monitoring Results

Table 2 shows strong separation between high-progress and low-progress episodes. A logistic probe on SAE features reaches **0.918 AUROC**, with 0.839 recall and 0.913 PR-AUC. Even a sparse top-20-feature probe remains competitive at 0.894 AUROC, confirming that progress information is concentrated in a compact subset of internal directions.

Table 1: Labeled dataset statistics for progress analysis.

Suite	Low-progress	High-progress	Total
goal	129	45	174
long	32	12	44
object	17	125	142
spatial	9	5	14
Low-progress timesteps	112200		
High-progress timesteps	112200		
Progress norm (low mean±std)	0.373 ± 0.172		
Progress norm (high mean±std)	0.952 ± 0.028		

Table 2: Overall progress classification performance (layer 20). Bold indicates best per column.

Method	AUROC↑	F1↑	Precision↑	Recall↑	PR-AUC↑
SAE Feature LR (16384-d)	**0.918**	**0.817**	0.797	**0.839**	**0.913**
Top-20 Feature LR	0.894	0.752	**0.844**	0.679	0.885
Random	0.554	0.550	0.566	0.536	0.563

Table 3: Motion-only control baselines vs. SAE-feature readouts.

Control Feature Set	AUROC↑
Action magnitude only	0.572
End-effector velocity only	0.711
Action magnitude + velocity	0.572
SAE Feature LR (16384-d)	**0.918**
Top-20 SAE Feature LR	0.894

Motion-magnitude control. To test whether classification performance is explained by trivial movement heuristics, we train control probes using only episode-level action magnitude and end-effector velocity summaries. Table 3 shows that these controls are substantially weaker than SAE-feature probes, supporting that learned separability is not reducible to gross motion magnitude alone.

Sparsity and information concentration. Figure 2 shows a steep sparsity-performance curve: AUROC rises rapidly with top-ranked features and saturates near full-model performance, indicating that progress-relevant signal is highly concentrated rather than uniformly distributed.

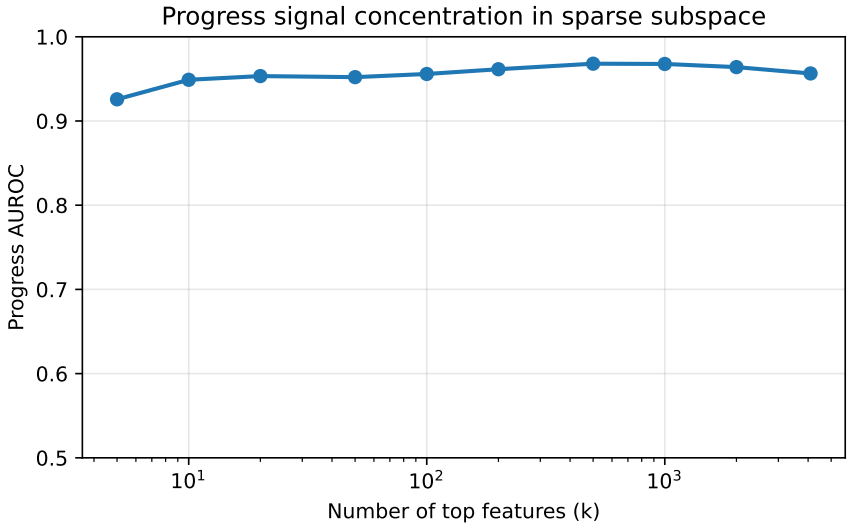

Fig. 2: Progress AUROC as a function of top-ranked feature count. Most separability is captured by a small subset of dimensions.

Table 4: Per-suite progress AUROC (layer 20).

Task Suite	Progress AUROC	Episodes	Interpretation
goal	**0.985**	174	Near-perfect progress separation
object	0.947	142	Strong and stable separation
long	0.700	44	Moderate, above chance
spatial	0.667	14	Positive but data-limited regime

4.3 Per-Suite Analysis

Table 4 reports per-suite progress AUROC. Performance is strongest on `goal` (0.985) and `object` (0.947), remains above chance on `long` (0.700), and is lower but positive on `spatial` (0.667), where the labeled sample is smallest.

4.4 Feature-Level Analysis

Differential analysis. Applying Mann–Whitney tests with BH-FDR correction ($\alpha = 0.05$) identifies **1,117 significant active SAE features out of 1,881 tested**. The volcano plot in Figure 3 shows a dense high-effect/high-significance region, indicating strong linear separability between high- and low-progress episodes.

Top features have large, stable effects. Table 5 lists the top-10 SAE features ranked by composite score. Effect magnitudes are substantial (rank-biserial $|r| \approx 0.22$–0.40) with extremely small adjusted p-values, supporting a robust and not merely marginal separation.

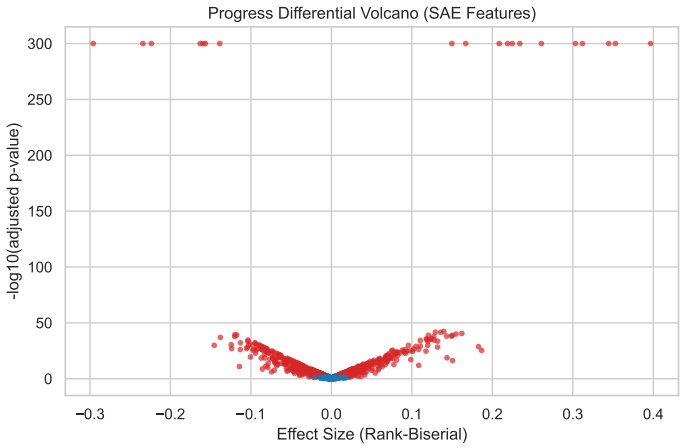

Fig. 3: Volcano plot of progress differential analysis at layer 20. Red points indicate dimensions passing FDR < 0.05.

Table 5: Top-10 progress-associated dimensions by composite score (layer 20).

Rank	Feature	Effect Size	Adj. p-value	Direction
1	10609	0.397	$< 10^{-300}$	higher in low progress
2	11471	0.353	$< 10^{-300}$	higher in low progress
3	1972	0.345	$< 10^{-300}$	higher in low progress
4	9215	0.312	$< 10^{-300}$	higher in low progress
5	11070	0.303	$< 10^{-300}$	higher in low progress
6	3154	-0.296	$< 10^{-300}$	higher in high progress
7	11183	0.261	$< 10^{-300}$	higher in low progress
8	9998	-0.234	$< 10^{-300}$	higher in high progress
9	15966	0.234	$< 10^{-300}$	higher in low progress
10	1528	0.225	$< 10^{-300}$	higher in low progress

Feature heatmaps. Figure 4 visualizes top-feature activations across episodes ordered by progress. The map exhibits clear stratification between low-progress and high-progress regions, reinforcing the differential ranking.

Stage-resolved feature behavior. Figure 5 reports normalized-trajectory activation profiles for top-ranked raw dimensions and representative trajectory-stage snapshots (early/mid/late). The profiles show consistent stage-dependent activation structure between low-progress and high-progress episodes, providing direct interpretability evidence beyond aggregate statistics.

4.5 Cross-Model Intervention Evidence

The following cross-model results are qualitative support that complements the direct on-OpenVLA interventions in Sec. 4.6, and we no longer present them as

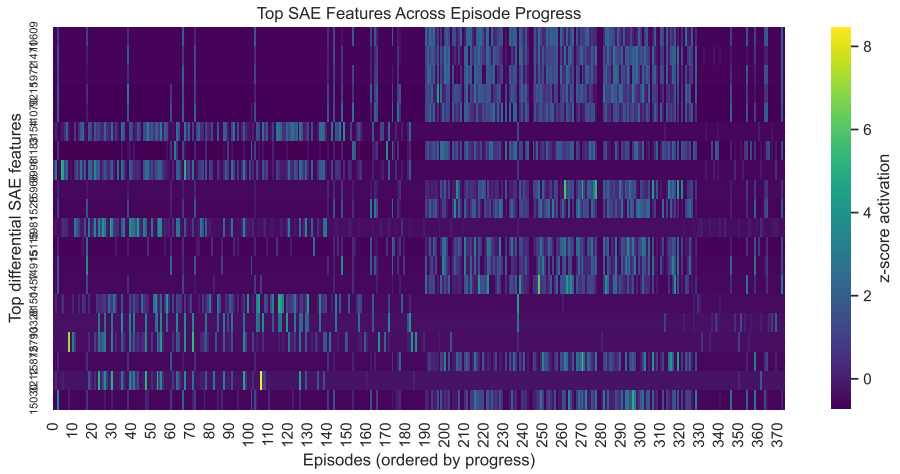

Fig. 4: Activation heatmap of top progress-associated features, with episodes ordered by progress score.

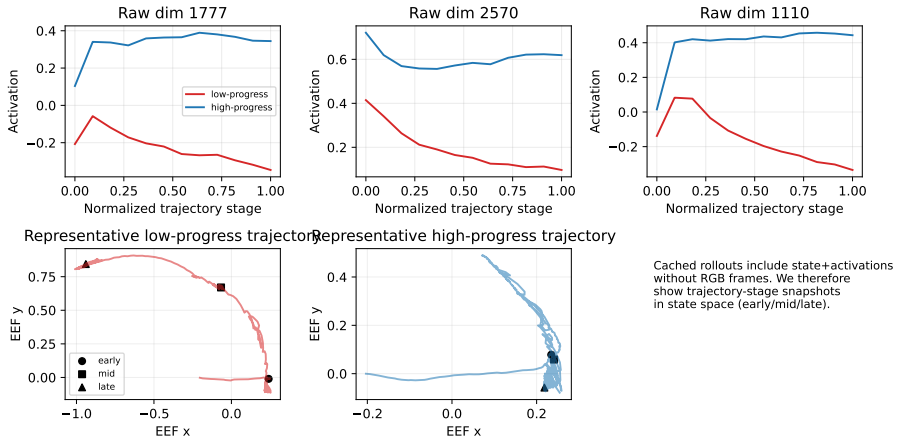

Fig. 5: Top-row: activation vs. normalized trajectory stage for top-ranked raw dimensions. Bottom-row: representative state-space trajectory snapshots (early/mid/late) for low-progress and high-progress episodes.

the primary causal evidence. To test whether sparse feature axes are actionable beyond OpenVLA, we include external intervention evidence from cross-model diffusion-policy experiments on Octo-small-1.5 in SimplerEnv. In these runs, baseline task success is low but non-zero (episode success 0.06 over 50 episodes, step success 0.000775), so success deltas are noisy and action-space metrics remain the primary signal.

Table 6: Cross-model intervention evidence from diffusion-policy interventions on Octo-small-1.5. Flips count axis-wise sign inversions between zero and amplify for each feature.

Feature	Mode	Norm	Block Δ	Δx	Δy	Δz	Flips
Baseline	–	0.103	–	–	–	–	–
F1383	zero	0.544	7.29	-0.028	+0.042	-0.031	3/3
F1383	amplify	0.362	4.49	+0.025	-0.064	+0.029	
F2347	zero	0.543	5.88	+0.010	-0.019	-0.016	2/3
F2347	amplify	0.434	1.02	-0.006	-0.017	+0.019	
F1982	zero	0.537	4.42	-0.001	-0.014	+0.011	1/3
F1982	amplify	0.494	2.91	+0.002	-0.021	+0.002	

Table 7: Live intervention evidence from run 325976 (eight complete zero-ablation feature blocks).

Feature ID	Episodes logged
F2250	12/12
F2371	12/12
F3708	12/12
F0637	12/12
F3231	12/12
F1982	12/12
F3256	12/12
F3988	12/12

Table 6 shows structured directional effects under zero vs. amplify interventions. Most notably, F1383 exhibits full XYZ sign inversion, while F2347 and F1982 show partial axis inversion. This provides external intervention evidence that top sparse directions correspond to intervention-sensitive control axes in a related VLA-family policy.

Live intervention run. We additionally include live ablation run 325976 as further live evidence. Parsing the logs yields eight complete zero-ablation feature blocks (Table 7). In this run, baseline mean action norm is 0.1045, while intervened mean norms range from 0.4596 to 0.5895 (4.40×–5.64× baseline), with per-feature XYZ deltas logged. This supports the same directional-control interpretation under live rollout conditions.

4.6 Baselines, Robustness, and Direct OpenVLA Intervention

We report dense baselines, robustness checks, a geometric-vs.-semantic audit, and direct on-OpenVLA interventions. Unless noted, these use a pooled all-episode layer-20 encoding with 2000-sample bootstrap 95% confidence intervals. Point estimates are consistent with the stratified protocol of Table 2.

Table 8: Progress probes on the layer-20 quartile split (pooled all-episode encoding, 2000-bootstrap 95% CIs). Dense and PCA baselines match or exceed the SAE, so the value of the sparse basis is inspectability and intervention compatibility rather than peak AUROC.

Probe (layer 20, same split)	AUROC (95% CI)
Raw activation LR	0.975 (0.960–0.987)
PCA-20	0.966 (0.947–0.982)
Raw activation MLP	0.957 (0.936–0.975)
Full SAE LR	0.947 (0.920–0.970)
Top-20 SAE LR	0.926 (0.900–0.951)
Random-projection-20	0.916 (0.883–0.945)
Motion telemetry	0.902 (0.870–0.933)
Random-20 SAE	0.783 (0.736–0.828)

Dense baselines and what the SAE adds. Table 8 compares progress probes on the same split. Raw-activation logistic regression is strongest (0.975) and PCA-20 (principal-component analysis, 0.966) matches the full SAE, so the SAE does not maximize AUROC by design. Its value is that nearly all the signal survives in 20 named, readable, writable directions. PCA-20 matches AUROC but is opaque, the raw MLP exposes no units, and a random 20-feature subset collapses to 0.783. The sparse basis is what enables the targeted readout and interventions below.

Layer choice. Progress is decodable across mid-to-late layers: monitor AUROC is 0.884 / 0.896 / 0.871 at layers 16 / 20 / 24 (bootstrap CIs ± 0.03), with top-feature Jaccard overlap ≤ 0.03 across layers. Layer 20 is best but not uniquely so, and the signal is not an artifact of one arbitrary layer choice.

Split robustness. The quartile discovery split is not load-bearing. Re-encoding all episodes and re-fitting, full-SAE AUROC is 0.948 under a tertile split and 0.929 under a median split. Aggregating statistics at the episode level (rather than per timestep) still yields 729 BH-FDR-significant features, confirming the signal is not an artifact of timestep pseudo-replication, and a 32K-dictionary SAE preserves it (full / top-20 0.950 / 0.938). The specific top-20 feature *identities* are tied to the quartile discovery setting, and we do not claim they transfer verbatim across split definitions.

Early-window diagnostics and transfer. Using only the first 25% of each rollout, the top-20 SAE / motion / motion+top-20 readouts reach 0.863 / 0.868 / 0.882 AUROC (TPR 0.85 at 20% FPR), indicating progress is partially decodable early enough for diagnostic use. Under a task-family shift (train on `goal`+`object`, test on `spatial`+`long`), the train-top-20 readout transfers to held-out tasks / instructions at 0.800 / 0.817, and the sparse top-20 outperforms the full dictionary (0.658 vs. 0.537), suggesting the ranked directions capture reusable rather than purely task-specific structure.

Confounds and uncertainty. All added numbers carry 2000-bootstrap 95% CIs. We demote the `spatial` suite (14 episodes, wide interval) and do not rely on it. To rule out suite identity, within-suite quartile splits (47 episodes/suite) give full / top-20 SAE 0.876 / 0.852 versus 0.431 for a suite-ID control, and within same-task, same-instruction pairs (n=1,357) the full / top-20 SAE rank higher above lower progress at 0.81 / 0.72. Separability is thus not explained by task or suite identity.

Geometric vs. semantic progress. Our target is *relative geometric progress*, and we verify what it is *not*. On 120 fresh OpenVLA rollouts on `goal`+`object` with real success/reward/object-state telemetry, the full SAE basis predicts episode success at 0.968 (95% CI [0.940, 0.990]), on par with raw activations (0.976). The progress-tuned top-20, by contrast, scores 0.471 on success, the correct null for a probe never trained on completion. A proxy audit confirms the target tracks final-distance telemetry ($\rho = -0.926$) but not episode-success metadata ($\rho = 0.012$). The task-state signal lives in the broader SAE basis, whereas the sparse top-20 is progress-specific, so we do not claim semantic completion.

Direct OpenVLA interventions. We complement the cross-model evidence of Sec. 4.5 with two direct on-OpenVLA checks. (i) Setting the submitted top-20 features to their high-progress class means shifts an *independent* raw-activation-trained progress readout by +0.763 (95% CI [0.65, 0.87]), versus +0.078 for matched-random features (permutation $p = 0.005$, 1000 trials). An offline action readout shifts 0.027 vs. 0.005 in L_2. Because the readout never sees SAE codes, this rules out circularity. (ii) In a pre-registered goal-task-1 specificity batch (n=64), setting the 13 active submitted features to class-mean reduces any-violation incidence from 1.00 to 0.50 ($\Delta = -0.50$, CI $[-0.625, -0.375]$, Wilcoxon $p = 1.5 \times 10^{-8}$), exceeding the strongest of seven pre-specified controls (three shuffled-value and four matched-random), whose best effect was $\Delta = -0.266$. Task success is unchanged, so this is task-local intervention sensitivity, not closed-loop repair.

5 Discussion

5.1 What Works: Strong Progress Signal in OpenVLA Internals

Our central finding is that OpenVLA SAE features carry a highly discriminative progress signal. The high-progress vs. low-progress split yields 0.918 AUROC with a linear probe, while the top-20 ranked features retain 0.894 AUROC. This gap between full-dimensional and sparse-top-k performance is modest, indicating that most separability mass is concentrated in a compact subset of internal directions.

The practical implication is compelling: lightweight linear readouts are already sufficient to recover success-related internal structure, enabling fast and interpretable diagnostics without policy retraining or additional data collection.

5.2 Robustness Across Task Suites

Per-suite AUROC remains high on `goal` (0.985) and `object` (0.947), moderate on `long` (0.700), and lower on `spatial` (0.667), where sample count is smallest. The ranking supports a robust core claim: progress-separating structure is present across suites, with expected degradation in low-data regimes.

5.3 Methodological Caveat

The progress signal relies on a *relative geometric progress* metric from end-effector displacement, as direct task-completion fields are absent in the shared metadata. We are explicit that this is geometric, not semantic (Sec. 4.6). The progress-tuned top-20 features score at chance on episode success, although the broader SAE basis does carry task-state signal. Still, the learned signal is not reducible to motion magnitude. Motion-only controls are far weaker (0.57–0.71 vs. 0.918), separability persists across suites, within suites, and within same-task pairs, and it is not attributable to suite identity. Future versions should incorporate explicit task-goal distance or completion metadata when available.

5.4 Broader Implication

The core takeaway is that OpenVLA internals contain a strong and sparse progress code in SAE space. This provides an interpretable ranking of internal directions, and we take a first step toward causal testing directly on OpenVLA: setting top-ranked features to their high-progress class means shifts an independent readout far beyond matched-random controls, and a pre-registered specificity batch produces task-local behavioral change without success repair (Sec. 4.6). These are feature-level intervention sanity checks, not closed-loop causal validation, which remains future work.

5.5 Cross-Model Evidence in Context

The primary intervention evidence is the direct on-OpenVLA checks of Sec. 4.6. As secondary, cross-model support, we compared this progress-centric finding to independent diffusion-policy interventions on Octo-small-1.5 in SimplerEnv. We emphasize that baseline task success in this setting is low but non-zero (episode success 0.06, step success 0.000775), so success-rate deltas are high-variance at this scale and should be read as qualitative. Accordingly, action-space metrics are the readout and show directional structure under feature interventions.

In particular, top feature F1383 shows a full XYZ sign inversion between ablation (zero) and amplification: $\Delta x = -0.028,\ \Delta y = +0.042,\ \Delta z = -0.031$ under zero versus $\Delta x = +0.025,\ \Delta y = -0.064,\ \Delta z = +0.029$ under amplify, with large intervention magnitude (block delta 7.29 vs. 4.49). This pattern is consistent with bidirectional steering of motion direction by a single sparse feature axis. Other top features (e.g., F2347, F1982) show partial axis flips, indicating heterogeneous but still structured causal influence. A separate live run (325976) produced eight

completed 12-episode intervention blocks, with 4.40×–5.64× action-norm amplification over baseline, reinforcing that this effect persists outside short controlled batches.

These external results do not replace the OpenVLA progress evidence, but they provide supportive intervention evidence that sparse ranked directions capture reusable behavioral structure across VLA-family policies.

6 Conclusion

We introduced a progress-based SafeSAE-VLA re-analysis that replaces a degenerate safe/unsafe split with a statistically meaningful high-progress vs. low-progress formulation. On 750 cached OpenVLA rollouts, this relabeling yields a strong and interpretable signal in internal activations.

Differential analysis identifies 1,117 significant active SAE features (FDR < 0.05) out of 1,881 tested at layer 20. A linear progress monitor reaches 0.918 AUROC overall, and top-20 ranked features still deliver 0.894 AUROC, demonstrating that progress information is both strong and sparsely concentrated. Dense and PCA baselines match or exceed these numbers, so the sparse basis serves as an interpretability and intervention interface rather than a stronger classifier, scoped to relative geometric progress.

These results provide a mechanistic foundation for sparse representation-based interpretation of VLA policies. As a first step toward causal grounding, direct on-OpenVLA feature-setting shifts an independent readout far beyond matched-random controls and produces task-local behavioral change in a pre-registered specificity batch, with cross-model diffusion-policy interventions offering secondary qualitative support. We do not claim closed-loop repair.

Future work. Natural next steps are (i) closed-loop causal validation, testing whether feature-setting interventions shift task *success* rather than only local action statistics; (ii) extending the analysis beyond OpenVLA and LIBERO to additional VLA families, benchmarks, and real-robot rollouts; and (iii) moving from relative geometric progress toward semantic progress by incorporating explicit task-completion or goal-distance metadata where available.

Acknowledgments. We thank the anonymous ECCV 2026 reviewers and the Area Chair, whose feedback prompted the dense-baseline comparisons, the layer and split-robustness analyses, the geometric-vs.-semantic audit, and the direct on-OpenVLA interventions added here.

Code and Data Availability. Code: `https://github.com/socratesosorio/safesae-vla-eccv2026`. Rollouts and layer-20 SAE checkpoint: `https://huggingface.co/datasets/socratesosorio/safesae-vla-eccv2026` (accessed 30 June 2026).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in AI safety. arXiv preprint arXiv:1606.06565 (2016)
2. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological) **57**(1), 289–300 (1995)
3. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
4. Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al.: Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread (2023), `https://transformer-circuits.pub/2023/monosemantic-features/index.html`, accessed 30 June 2026
5. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gober, K., Hausman, K., Herzog, A., Jasber, J., et al.: RT-1: Robotics transformer for real-world control at scale. In: Robotics: Science and Systems (RSS) (2023)
6. Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems **5**, 411–444 (2022)
7. Conmy, A., Mavor-Parker, A.N., Lynch, A., Heimersheim, S., Garriga-Alonso, A.: Towards automated circuit discovery for mechanistic interpretability. Advances in Neural Information Processing Systems **36** (2023)
8. Cunningham, H., Ewart, A., Riggs, L., Huben, R., Sharkey, L.: Sparse autoencoders find highly interpretable features in language models. arXiv preprint arXiv:2309.08600 (2023)
9. Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., Tassa, Y.: Safe exploration in continuous action spaces. arXiv preprint arXiv:1801.08757 (2018)
10. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: PaLM-E: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
11. Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al.: Toy models of superposition. Transformer Circuits Thread (2022), `https://transformer-circuits.pub/2022/toy_model/index.html`, accessed 30 June 2026
12. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research **16**(42), 1437–1480 (2015)
13. Hendrycks, D., Carlini, N., Schulman, J., Steinhardt, J.: Unsolved problems in ML safety. arXiv preprint arXiv:2109.13916 (2021)
14. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: OpenVLA: An open-source vision-language-action model. In: Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 2679–2713 (2025), `https://proceedings.mlr.press/v270/kim25c.html`, accessed 30 June 2026
15. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2023), `https://proceedings.neurips.`

cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html, accessed 30 June 2026
16. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The Annals of Mathematical Statistics **18**(1), 50–60 (1947)
17. Octo Model Team, Ghosh, D., Walke, H.R., Pertsch, K., Black, K., Mees, O., et al.: Octo: An open-source generalist robot policy. In: Robotics: Science and Systems (RSS) (2024), https://roboticsconference.org/2024/program/papers/90/, accessed 30 June 2026
18. Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., Carter, S.: Zoom in: An introduction to circuits. Distill (2020). https://doi.org/10.23915/distill.00024.001
19. Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., et al.: Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Transformer Circuits Thread (2024), https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html, accessed 30 June 2026
20. Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J.E., Ibarz, J., Finn, C., Goldberg, K.: Recovery RL: Safe reinforcement learning with learned recovery zones. IEEE Robotics and Automation Letters **6**(3), 4915–4922 (2021)
21. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Proceedings of The 7th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 229, pp. 2165–2183 (2023), https://proceedings.mlr.press/v229/zitkovich23a.html, accessed 30 June 2026