You Cannot Monitor Your Way Out of a Design Deficit: Inspectable VLA Internals as a Shared Safety Substrate

Socrates Osorio*
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, USA
Email: socratesj.osorio@berkeley.edu

Joy Zheyun Yang*
University of Oxford
Oxford, United Kingdom
Email: joy@robots.ox.ac.uk

***Abstract*—Safety for embodied foundation models is increasingly organized around two paradigms, safety *by design* versus safety *by practice*, usually posed as a choice between two toolboxes operating on two surfaces: design shapes the training pipeline, while practice wraps a finished black box at deployment. We take a position against this dichotomy and argue, with measurements rather than a new algorithm, that for vision-language-action (VLA) foundation models the two paradigms are a *dependency* mediated by a single object: the policy's own internal representations. We support the position with three evidence-backed claims read off a sparse-autoencoder (SAE) basis over OpenVLA residual streams. (1) Practice has a floor that design sets. Our first benchmark was statistically degenerate: a naive binary safe/unsafe target had near-universal violations and almost no variance, so no runtime monitor could have learned it: you cannot monitor your way out of a design deficit. (2) The interior is the bridge. The same inspectable basis is simultaneously a practice-time monitor and a design-time audit of *what the model represents*: task progress is strongly, sparsely, and *causally* encoded (AUROC 0.918, with $1{,}117$ FDR-significant features and an intervention logit shift of 0.763 vs. 0.078 for matched random, $p{=}0.005$), whereas hazard is encoded only weakly and diffusely at the action-token interface we read, a representational gap that no deployment-time wrapper reading that interface can recover. What a model does not represent where you read it, you cannot monitor. (3) Select practice-time monitors on calibration, not accuracy. On a telemetry-audited 560-rollout prefix-only future-violation benchmark, an SAE monitor and a raw-activation MLP are statistically tied on AUROC (0.632 vs. 0.640), but after identical calibration the inspectable monitor is $2.4\times$ better calibrated (ECE 0.143 vs. 0.339), is the only monitor above chance on all five hazard categories, and runs at 1.18 ms/step, so its confidences are usable under an abstention or cost-weighted policy while an equally accurate black box's are not. A cross-architecture Octo-small-1.5 intervention shows the substrate is not OpenVLA-specific. *Scope:* all results are in simulation with sim-audit-defined thresholds, the audit reads action-token residual streams, so weak hazard coding there does not rule out hazard structure in earlier vision-language layers, and the π_0 and real-robot transfers are open. The contribution is a position and a concrete toolbox proposal: build embodied-FM safety on an inspectable internal substrate that lets design and practice read, and act on, the same evidence.**

*Equal contribution.

I. Introduction

Vision-language-action (VLA) foundation models map language and pixels directly to low-level robot actions [28, 18, 3], and are moving quickly from benchmarks into closed-loop deployment. Their safety is commonly framed along two complementary axes: *safety by design*, which reduces risk at the source through data, architecture, and objectives, and *safety by practice*, which manages risk at deployment through monitoring, red-teaming, shielding, and recovery. The two are typically treated as separate communities with separate surfaces of action: design owns the training pipeline, practice owns a finished, opaque controller.

Position. We argue this separation is misleading for embodied foundation models, and that the cleaner picture is a *dependency* mediated by one object: the policy's internal representations. Concretely, (i) safety by practice has a measurement *floor* that safety by design must clear before any monitor is even learnable, (ii) the policy's interior is a single substrate that both *audits the design* (it reveals which safety-relevant concepts a model does and does not encode) and *instruments the practice* (it supplies a runtime monitor), and (iii) the property a practice-time monitor should be selected on is *calibration*, not ranking accuracy. We are not proposing a new training algorithm or a new shield. We are proposing a substrate, and a way to choose tools on it, backed by measurements on a real 7B VLA.

We instantiate the substrate with sparse autoencoders (SAEs) [4, 10, 26], which decompose a model's residual stream (the running hidden vector that each transformer layer reads and writes) into a basis of sparse, individually inspectable features. SAEs have illuminated language-model internals. We bring them to a VLA foundation model and treat its residual stream as a visual decision backbone whose internal progress and risk code can be read, ranked, attributed, and perturbed: by a designer auditing the model and by a monitor watching it at runtime.

An origin lesson. Our first rollout set exposed the dependency directly. A naive binary safe/unsafe label was statis-

tically *degenerate*: violations were near-universal, leaving almost no discriminative variance for a monitor to latch onto. No amount of practice-time cleverness recovers a signal that the policy's behavior never produced. We therefore adopt a suite-normalized high-vs.-low *relative geometric progress* target as a clean, variance-preserving probe of internal structure, and treat safety as a separate, harder stress test on a telemetry-audited benchmark. This is the position in miniature: the monitorability of a deployed policy is bounded by what its design left behind.

Contributions. As a position paper grounded in measurement, we contribute:

1) **A reframing of the design-versus-practice dichotomy** as a dependency, with the inspectable interior as the shared substrate where design-time auditing and practice-time monitoring read the same evidence (Sec. II).
2) **A design-readable representational audit.** On OpenVLA, progress is strongly, sparsely, and causally encoded (0.918 AUROC, 0.894 from $\sim$20 directions, $1{,}117$ FDR-significant features, causal logit shift 0.763 vs. 0.078), while hazard is encoded only weakly and diffusely at the action-token interface, a gap a designer can read but a monitor at that interface cannot fill (Sec. III).
3) **A selection criterion for practice-time monitors.** Two AUROC-tied violation monitors differ $2.4\times$ in calibration error, and the well-calibrated, 1.18 ms/step one reads off the inspectable basis. Embodied-FM monitors should be chosen on calibration, not AUROC (Sec. III, Table III).
4) **A cross-architecture datapoint and an honest scope.** The same intervention recipe steers Octo-small-1.5. All results are in simulation, and we name the open transfers (Sec. IV).

II. Position: Design and Practice Meet in the Interior

a) Claim 1: Practice has a floor that design sets.: A runtime monitor, a red-team probe, or a shield is a function of the policy's behavior and internal state. If design produced a policy whose behavior is so uniformly unsafe (or so uniformly safe) that the hazard label has no variance, there is nothing for any practice-time tool to learn. This is not a tuning problem but an information-theoretic one. Our degenerate first benchmark is a concrete instance: with violations near-universal, a perfectly engineered monitor and a coin flip are indistinguishable. The corollary is uncomfortable for a deployment-only safety stance: *you cannot monitor your way out of a design deficit*. Practice-time assurance therefore cannot be evaluated in isolation, because its achievable performance is conditioned on what design left in the policy's behavior and representations.

b) Claim 2: The interior is the bridge, and what is not represented where you read it cannot be monitored.: Safety by design asks what a model encodes, and safety by practice asks what a deployed monitor can read. An inspectable internal basis answers both with one object. Reading it as a *design audit*, we find OpenVLA strongly and sparsely represents task progress but only weakly and diffusely represents hazard at the action-token positions we read (Sec. III). That asymmetry is a design finding (the model was never trained with an objective that concentrates hazard structure), and it directly bounds practice: a diffuse, weak internal hazard code is exactly why our best honest violation monitor is an *early* 0.632-AUROC signal, not a usable stand-alone alarm. Whether hazard information is absent from the network altogether, or present in earlier vision-language layers and lost before action generation, is a question the same substrate can answer, and either answer bounds a monitor reading the action interface. The actionable consequence is to move a representational target *upstream*, into design, rather than asking practice to manufacture a signal the readable interior does not contain.

c) Claim 3: Select practice-time monitors on calibration, not accuracy.: The runtime-assurance literature reports monitors by ranking accuracy (AUROC). But a deployed monitor must commit to an operating point, abstain under uncertainty, or feed a cost-weighted controller, all of which depend on whether the emitted *probability*, not merely its rank order, is trustworthy. We show two monitors tied on AUROC differ $2.4\times$ in calibration error (Sec. III). Calibration is the decision-relevant property, and an inspectable basis lets you obtain it *while* keeping the monitor auditable, so a flagged risk is attributed to named features, not emitted as an opaque scalar.

These three claims are a toolbox proposal, not three disconnected observations: build embodied-FM safety on an inspectable internal substrate, push representational targets into design where practice's floor is set, and select runtime tools on calibration. The rest of the paper is the evidence that makes this concrete and defensible today.

d) Related work: two literatures, one missing bridge.: Safety *by design* for robot policies descends from constrained and safe reinforcement learning [11, 15, 6] and the broader program of building risk out at the source [1, 17]. Safety *by practice* descends from deployment-time monitoring and recovery: learned recovery policies [27], system-level out-of-distribution and competence monitoring [24], and conformal failure prediction for vision-based control [14]. These monitors are effective but *black-box*: they score risk without exposing which internal features drive it. Mechanistic interpretability supplies the missing inspectability: circuit analysis [23, 9], superposition [13], and sparse dictionaries that decompose activations into inspectable features [4, 10, 26]. For VLA foundation models [5, 28, 18, 3, 22, 12], a young line probes representations with linear readouts [20, 8] and finds steerable sparse directions [16, 25]. Our position connects these threads: an inspectable basis is precisely the substrate on which the design and practice literatures can read, and act on, the same evidence.

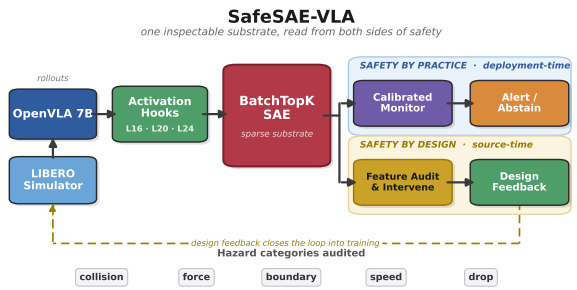

Fig. 1: The shared substrate. OpenVLA rollouts are relabeled for progress and safety, action-token residual activations are encoded by a trained BatchTopK SAE, and the same sparse basis feeds a design-time representational audit (differential analysis, intervention) and a practice-time runtime monitor (calibrated progress/violation readouts).

III. Evidence: An Inspectable Substrate on a 7B VLA

a) Substrate and pipeline.: We hook the OpenVLA [18] backbone and cache residual-stream activations $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^{4096}$ at action-token positions for layers $\ell \in \{16, 20, 24\}$ over 750 LIBERO episodes [19] (`spatial`, `object`, `goal`, `long`). We train BatchTopK SAEs [7] with d_{sae}=16384 (4× overcomplete), top-k=32, for 50M tokens, and the layer-20 model uses 1,881 of 16,384 features (Fig. 1). For progress we score each episode from simulator telemetry, normalize within suite, and take top vs. bottom quartile (374 balanced episodes). For safety we use a telemetry-audited benchmark with per-timestep thresholds for five hazard categories (collision, force, boundary, speed, object-drop), fixed by audit. Monitors are evaluated *prefix-only* (at step t, predict a violation within h=25 steps) under leave-one-task-out splits, so no hindsight leaks.

b) Design audit: progress is strong, sparse, and causal.: Reading the basis as a design audit (does the model encode progress in inspectable form?), the answer is decisively yes. A logistic probe on SAE features reaches 0.918 AUROC, and a top-20-feature probe retains 0.894, so most separability lives in a compact direction set (Table I). A per-feature Mann–Whitney test [21] with Benjamini–Hochberg FDR control [2] flags 1,117 of 1,881 active features as significant (α=0.05, Fig. 2 left). These directions are *causal*, not merely correlated: on held-out low-progress episodes, patching the top-20 features to their high-progress means raises the progress logit by 0.763 ($[0.653, 0.868]$) vs. 0.078 for 200 matched random feature sets (p=0.005, Fig. 2 right), with a monotone dose-response. Motion-only controls reach just 0.57–0.71 AUROC, so this is not gross movement magnitude.

c) Design audit: hazard is weak and diffuse, the bridge claim made concrete.: The same basis read for hazard tells the opposite story, and that asymmetry is the point. The top-20 *geometric* progress features do not predict task success (0.471) or object-lift (0.442): they track motion, not completion (Table II), an honest scope boundary, not a failure. When labels exist, the *full* basis does carry strong outcome signal (0.968 AUROC on true success, near raw activations),

TABLE I: Design audit: progress classification (layer 20). **All AUROCs target *relative geometric progress*, a distance proxy, not task success**: the same top-20 features score only 0.471 on true success (full basis 0.968, Table II), so 0.918 must not be read as a success detector. The audit answer is that progress is represented in a compact, inspectable code.

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 II: True-success / object-lift readout (120 success-labeled rollouts, Hanley–McNeil CIs). The full inspectable basis predicts real success near raw activations, while the deliberately narrow geometric features do not. The representation is rich, but hazard is not concentrated in it.

Readout	Success AUROC	Object-lift AUROC	95% CI (succ.)
Full SAE LR	**0.968**	**0.964**	[0.940, 0.990]
Full SAE PCA-20	0.949	0.949	[0.910, 0.979]
Raw Act. LR	0.976	0.991	[0.952, 0.993]
Geometry-only	0.665	0.549	–
Top-20 geom. feat.	0.471	0.442	–

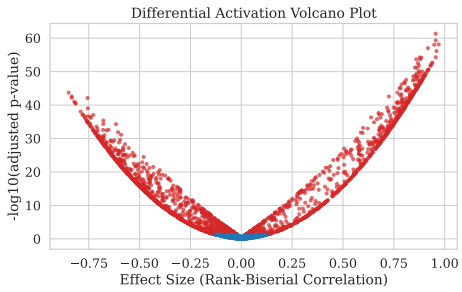

Fig. 2: Volcano plot of the progress differential audit (layer 20), with red marking FDR $<$ 0.05 (1,117 of 1,881 active features). These directions are causal: patching the top-20 to their high-progress means shifts the progress logit by 0.763 vs. 0.078 for matched random features (p=0.005).

so the representation is rich, but hazard specifically is not concentrated. Progress and safety features are nearly disjoint (top-50 Jaccard 0.020), and the prefix-only violation signal is broad-but-weak. A designer reads this as: at the interface where actions are generated, the model encodes "am I moving toward the goal" far more cleanly than "am I about to do harm." This audit is scoped to action-token positions: it establishes that hazard is not preserved where actions are read out, not that it is absent from the network, and repeating it on earlier vision-language layers is the direct next probe. *What the model does not represent where the monitor reads, the monitor cannot recover*, which is why the next result is framed as a selection criterion, not an accuracy win.

d) Practice tooling: calibration, not accuracy, separates two tied monitors.: We stress-test the basis on the harder prefix-only future-violation setting (Table III). Aggregate AUROC is *not* a win: the raw-activation MLP (0.640) ties the SAE readout (0.632, 7 folds, 560 rollouts) within fold-to-fold variation, so we claim no accuracy edge and report 0.632

TABLE III: Practice-time monitor selection: prefix-only future-violation monitoring (telemetry-audited, 560 rollouts, leave-one-task-out, 7 folds, horizon 25). AUROC is threshold-free, and ECE/Brier are after identical Platt calibration (lower is better). The inspectable SAE monitor *ties* the raw MLP on AUROC but cuts calibration error $2.4\times$, the property an operating-point/abstention policy depends on.

Method	AUROC↑	PR-AUC↑	ECE↓	Brier↓	Med. Lead
Raw Activation MLP	**0.640**	**0.535**	0.339	0.362	51.7
SAE Feature LR	0.632	0.470	**0.143**	**0.232**	47.6
Force Threshold	0.615	0.477	0.170	0.235	40.0
Raw Activation LR	0.577	–	0.383	0.404	64.5
Random	0.500	0.327	0.176	0.250	51.2

honestly as an early signal. The selection-relevant result is that, after identical Platt calibration, the inspectable monitor is decisively and stably better calibrated: ECE 0.143 ± 0.061 vs. 0.339 ± 0.203, Brier 0.232 vs. 0.362, $2.4\times$ lower calibration error at $3.4\times$ lower fold variance. It is the *only* monitor above chance on all five hazard categories, and it runs at 1.18 ms/step (App. N), inside a 5–10 Hz control budget. The same-classifier comparison (SAE LR vs. raw LR) isolates the representation and still favors the inspectable basis (ECE 0.143 vs. 0.383). Concretely: when the SAE monitor outputs $p{=}0.3$ a violation follows $\approx 30\%$ of the time, so an operator's abstention band or cost-weighted threshold can take its confidence at face value, while the equally accurate black box's cannot.

e) Auditability is the bridge in action.: Because the monitor reads named features, a flagged rollout is legible. On a stalled `long` episode (rollout 371), high-progress features that fire densely on a matched success (62–67%) are absent throughout the stall, so an operator reads off *which* progress concepts failed to engage and *when* (App. D), attribution a scalar alarm cannot give, and exactly the information a designer needs to close the loop back into training.

f) The substrate is not OpenVLA-specific.: So the position does not rest on one model, we apply the same intervention recipe to Octo-small-1.5 [22], a diffusion-policy VLA in SimplerEnv. Single sparse features act as control axes there too: F1383 shows full XYZ sign inversion under zero-vs.-amplify ($3/3$ flips), and a live rollout drives mean action norm to 4.4–$5.6\times$ baseline (App. F). Octo's low SimplerEnv success (≈ 0.06) makes this a directional-control datapoint, not a matched progress-AUROC, and we flag the same-target π_0 replication as the highest-value next step.

IV. Discussion: A Toolbox for the Community

a) What the position buys.: Reading the dichotomy as a dependency yields a concrete research agenda rather than a slogan. (i) *Design-time representational targets.* Because what is not represented where monitors read cannot be monitored, training objectives should be measured not only on task success but on whether they concentrate hazard-relevant structure into an inspectable, monitorable code: an SAE audit makes this an observable design metric, and a concrete remedy is to fine-tune with an auxiliary hazard-prediction objective on telemetry-audited rollouts and accept a checkpoint only when the audit shows hazard structure concentrating at the layers monitors read. (ii) *Calibration-first practice.* Runtime-assurance benchmarks for embodied FMs should report calibration (ECE/Brier after matched recalibration), not AUROC alone, and should extend the comparison to temporal monitors (recurrent or attention heads over feature histories) under the same protocol. Our result shows that accuracy and calibration can disagree on which monitor to ship. (iii) *Steering as a design intervention.* Our interventions causally move the progress logit and, on Octo, the action stream: the same handles that make a monitor auditable are candidate shielding/edit handles, blurring the practice/design line by construction. The unifying recommendation is to make embodied-FM safety *read* the policy rather than only *wrap* it.

b) Scope and limitations.: We are deliberate about what is *not* shown. (i) **No real-robot validation**: thresholds are sim-audit-defined, bounded only by a $\pm20\%$ sweep (App. K). (ii) The 0.632 AUROC is an *early* signal, not a stand-alone alarm. The calibration *ranking*, and the design lesson behind it, are the contributions. (iii) The audit reads action-token residual streams only, so "weak and diffuse" is a claim about the interface monitors read, not the whole network. Auditing earlier vision-language representations to distinguish absence from loss is the direct next probe. (iv) Cross-VLA evidence is intervention-only (Octo directional control), OpenVLA and Octo predate current flow-matching VLAs, and the π_0 progress-AUROC replication is unrun and highest-value. (v) The top-20 target is geometric progress (0.471 on success, full basis 0.968). (vi) Closed-loop behavioral causality is unestablished ($n{=}12$, underpowered, App. E). None of these undercut the position: each is a place where design and practice must jointly close a loop the interior has made visible.

c) Conclusion.: Safety by design versus safety by practice is a false choice for embodied foundation models. Practice has a floor that design sets, the policy's inspectable interior audits the design and instruments the practice with one substrate, and the right way to select a runtime monitor is calibration, not accuracy. You cannot monitor your way out of a design deficit, but you can read the policy to find out where the deficit is, and act on the same evidence from both sides. The next steps are an audit of earlier vision-language layers to locate where hazard information is lost, a powered π_0 replication, and real-robot transfer.

Acknowledgments

We thank the reviewers and program chairs of the Trustworthy Embodied Foundation Models workshop. Their feedback shaped the camera-ready scoping of the representational audit to the action-token interface, the explicit distinction between hazard information being absent and being lost before action generation, and the sharper statement of the design-time remedy the position proposes.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

[4] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gober, Karol Hausman, Alexander Herzog, Jasmine Jasber, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

[6] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. 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] Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

[8] Hugo Buurmeijer, Carmen Amo Alonso, Aiden Swann, and Marco Pavone. Observing and controlling features in vision-language-action models. *arXiv preprint arXiv:2603.05487*, 2026.

[9] Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 2023.

[10] 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.

[11] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

[12] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[13] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

[14] Alec Farid, David Snyder, Allen Z. Ren, and Anirudha Majumdar. Failure prediction with statistical guarantees for vision-based robot control. In *Robotics: Science and Systems (RSS)*, 2022.

[15] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.

[16] Bear Häon, Kaylene Caswell 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, 2025. URL https://proceedings.mlr.press/v305/haon25a.html.

[17] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *arXiv preprint arXiv:2109.13916*, 2021.

[18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[19] 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 (NeurIPS)*, 2023.

[20] Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture. *arXiv preprint arXiv:2502.04558*, 2025.

[21] Henry B. Mann and Donald R. Whitney. 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.

[22] Octo Model Team, Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024. URL https://roboticsconference.org/2024/program/papers/90/.

[23] 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.

[24] Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M. Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2023.

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

[26] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[27] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[28] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. 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, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX: SUPPLEMENTARY ANALYSES

This appendix begins *after* the References and collects the auditability case study, closed-loop steering breakdown, dictionary-scaling and per-layer differential analyses, per-fold safety statistics, the matched-calibration false-alarm retraction, per-task breakdowns, and stress tests that did not fit the four-page body.

A. Labeled Dataset Statistics

Table IV reports per-suite low/high-progress counts and progress-norm statistics for the suite-normalized quartile split (374 labeled episodes, 187/187 balanced, 112,200 timesteps per class).

TABLE IV: Labeled dataset statistics for the progress audit.

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		

B. Sparsity–Performance Curve and Cross-Validated ROC

Fig. 3 expands the body claim that "most separability mass lives in a compact set of directions." Left: progress AUROC rises steeply and saturates by ∼20 features, so the top-20 probe (0.894) is within noise of the full-basis probe (0.918). Right: cross-validated ROC for full-basis (0.920) and nested top-20 (0.899) probes nearly coincide, so the compact code is not a single-split feature-selection artifact.

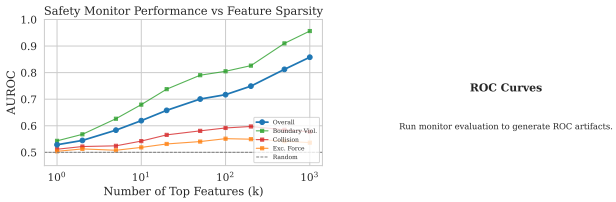

Fig. 3: Left: progress AUROC vs. number of retained top-ranked SAE features. Right: cross-validated ROC for full-basis and nested top-20 probes nearly coincide.

C. Leakage-Controlled Top-20 (Nested Selection)

Selecting the top-20 features inside each training fold only (no test-fold leakage) gives AUROC 0.874 $[0.837, 0.908]$, PR-AUC 0.864. The sparse-code story survives train-only selection. The selected features differ from any fixed submitted list, so the claim is "a compact code exists," not "these exact 20 directions are canonical."

D. Operator-Facing Failure-Attribution Case Study

Fig. 4 tracks three top-ranked progress features along normalized trajectory stage for a matched successful (`object`, rollout 575, final progress 0.95) and stalled (`long`, rollout 371, final progress 0.00) episode: high-progress features ramp and stay high on the success but never engage on the stall, while low-progress features invert. Because each plotted feature carries an FDR-controlled signature and a monotone dose-response (App. M), the attribution is not anecdotal: these are the same features whose causal patching shifts the progress logit.

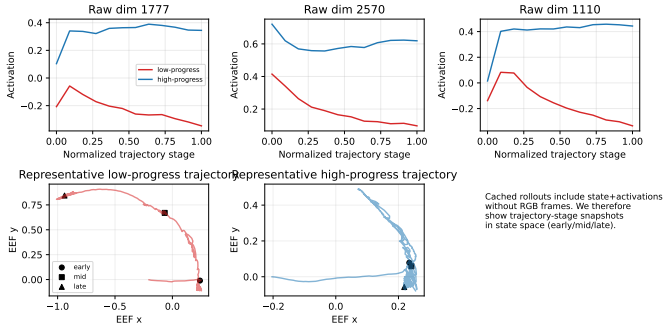

Fig. 4: Failure-attribution case study. High-progress features (blue) engage on a successful episode but stay flat on a stalled one (red), while low-progress features invert. Ranked features give an attributable account of *why* a rollout stalled.

E. Closed-Loop Directional Steering: Per-Direction Breakdown

Table V reports the n=12 paired closed-loop steering experiment (intervention latched from step 20). The harmful direction perturbs actions in 10/12 episodes and increases collisions (0→3/12) and mean violations/episode (1.00→1.58), while the helpful direction does not reliably increase success. Wilcoxon tests are underpowered at this n (collision p=0.25, success p=1.0), and a power analysis targets N≈40 paired rollouts for the follow-up. We report this as illustrative, not as runtime-control evidence.

TABLE V: Closed-loop directional steering (n=12 paired rollouts, layer 20, top-10 directional features, latched at $t \geq 20$).

Metric	Baseline	Helpful dir.	Harmful dir.
Collision rate	0.000	0.083	**0.250**
Any-violation rate	0.667	0.667	0.750
Success rate	0.583	0.500	0.583
Mean violations/episode	1.000	1.167	**1.583**
Action-delta $\neq 0$ rate	–	0.833	0.833
Wilcoxon p (collision)	–	1.00	0.25

F. Cross-Architecture (Octo-small-1.5) Intervention Detail

Zero-vs.-amplify SAE interventions on Octo-small-1.5 in SimplerEnv (a diffusion-policy VLA architecturally distinct from OpenVLA): F1383 produces a full 3/3 axis-wise sign inversion of the per-step XYZ action delta (block action-norm 0.103→0.544 under zero, 0.362 under amplify), F2347

a $2/3$ inversion, F1982 a $1/3$ inversion. A live ablation rollout yields eight complete zero-ablation feature blocks (each $12/12$ episodes) with intervened mean action norms 0.46–0.59 (4.4–$5.6\times$ the 0.105 baseline). *Scope caveat:* Octo's SimplerEnv baseline success is ≈ 0.06, too low for outcome metrics, so this is strictly a directional-control datapoint that the substrate is intervention-sensitive on a second architecture. The same-target π_0 progress-AUROC replication remains the highest-value next experiment.

G. Dictionary-Size Scaling (16K vs. 32K)

A wider layer-20 dictionary (d=32768, k=48) preserves and slightly strengthens the progress signal on both full and nested top-20 probes (0.950/0.938 AUROC) and surfaces more FDR-significant features, so the result is not an SAE-width artifact (sampled-episode sanity check: the main $1{,}117$/$1{,}881$ figure is the full analysis).

H. Per-Layer Differential Analysis and FVU

Layer 20 gives the best monitor AUROC (0.896 vs. 0.884/0.871 for L16/L24), and the three layers share very few significant features (Jaccard ≤ 0.03), carrying complementary progress information. Earlier layers reconstruct better (lower FVU) but monitor slightly worse, suggesting a multi-layer fusion opportunity left to future work.

I. Per-Fold Prefix-Safety Statistics

The SAE monitor's per-fold AUROC spans 0.58–0.69 (mean 0.633 ± 0.041 over 7 leave-one-task-out folds). The 0.632 vs. 0.640 gap to the raw MLP is smaller than this fold-to-fold variation, so the two are statistically indistinguishable in accuracy, which is why calibration, not AUROC, is the deciding property.

J. Matched-Calibration Safety (False-Alarm Retraction)

At each monitor's natural 0.5 threshold the SAE monitor shows a far lower false-alarm rate (0.009 vs. the raw MLP's 0.079), but this is an operating-point effect: under one identical Platt-scaling-plus-shared-false-alarm-constraint procedure the false-alarm rates converge (0.092 vs. 0.092), so *we retract the false-alarm edge*. The surviving, recall-matched edge is calibration: ECE 0.143/Brier 0.232 (SAE) vs. 0.339/0.362 (raw MLP) and 0.383/0.404 (raw LR), a 2.4–$2.7\times$ reduction. This is the assurance property an inspectable basis provides over an equally accurate black box.

K. Bounding the Sim-to-Real Threshold Gap

Because the five hazard thresholds are audit-defined, we bound the sim-to-real gap by rescaling every threshold $\pm10/20\%$, re-deriving ground truth, and recomputing monitor–label agreement. Category consistency stays ≥ 0.72 across the realistic 0.9–$1.2\times$ band and the monitor remains above chance throughout, degrading only at the $0.8\times$ extreme where boundary/speed labels become dense. This is a bounded claim, not a transfer guarantee, and real-robot recalibration of thresholds and the progress code remains required.

L. Seed Stability and Progress–Safety Disentanglement

A second SAE (seed 123) gives top-50 Jaccard 0.316 and Spearman ρ=0.675 on composite scores: feature identities shift but ranking structure and AUROC are preserved. Top-50 progress vs. top-50 safety features share only 2 (Jaccard 0.020): progress and safety occupy largely distinct subspaces, the representational asymmetry behind the bridge claim.

M. Dose-Response and Action-Readout Proxy

A raw hidden-state action readout reaches R^2=0.804, so decoded feature deltas translate to action-shift proxies. Amplifying the top-5 progress features produces monotonic hidden-state and action-readout shifts (Spearman ρ=1.0, p=1.4×10^{-24}), while bottom-ranked active features have near-zero effect, the monotonicity that licenses reading the case-study features causally.

N. Compute and Runtime Cost

The end-to-end monitor (SAE feature encode + calibrated logistic readout) costs 1.18 ms/step mean (1.23 ms p95): 0.89 ms encode + 0.29 ms readout, comfortably inside a 5–10 Hz VLA control budget. SAE training (50M tokens) is a one-time offline cost, and the monitors are L_2 logistic regressions requiring no policy retraining.