Extended Abstract

Motivation Vision-language-action (VLA) policies can execute diverse robot manipulation tasks, but under distribution shift they often fail silently: the policy drifts toward an unrecoverable state before the final failed action is visible. This project asks whether a frozen VLA policy can be made safer without updating its weights. We study a stricter inference-time setting: the base SmolVLA policy is held fixed, and a separate shield reads internal activations, predicts near-term failure risk, and conditionally injects a small activation-space correction.

Method Our system, SHIELD / FAPS, has three components. First, a risk head reads multi-layer SmolVLA action-expert activations and predicts binary horizon risk, time-to-failure (TTF), and a latent representation z_t. Second, a calibrated hysteresis gate decides when intervention is justified using a high threshold, a low exit threshold, and persistence. Third, a flow-matching residual model uses z_t to generate a low-rank hidden-state correction $\Delta h_k = U_k z$, where U_k is a PCA basis for transformer layer k. The base VLA weights are never updated.

Implementation We evaluate on LIBERO manipulation rollouts under four perturbation types: action noise, image blur, viewpoint changes, and joint noise. PyTorch hooks capture SmolVLA activations from six action-expert layers during rollout, producing a combined dataset of 5,117 nominal and perturbed training episodes with binary failure labels at horizon $H = 30$. Phase 1 trains and calibrates the risk head on this dataset using multi-objective supervision (BCE, TTF, NT-Xent) and temperature scaling; Phase 2 trains activation residuals using conceptor-derived subspace targets and a flow-matching objective; Phase 3 deploys the full shield in closed-loop LIBERO simulation and measures per-task success rate and false-positive rate against a matched unshielded baseline.

Results All three components produce positive offline signals. Failure risk is strongly readable from frozen activations: supervised risk heads reach roughly 0.92 weighted AUC, and our selected risk head used in Phase 3 achieves 0.911 wAUC with improved calibration (ECE 0.043). Contrastive supervision yields better failure/recovery geometry in the risk embedding, which downstream flow conditioning requires. Phase 2 confirms that conceptor-guided flow models learn nontrivial residuals ($\|\Delta\| \approx 2.0$), while unguided controls produce near-zero corrections, confirming the learned structure is necessary. In closed-loop Phase 3 evaluation, shielding produces measurable rescue under action noise—an action-expert-domain perturbation whose damage enters within the action expert's input space—with task-level success rate gains of up to $+0.50$. The key Phase 3 finding is a boundary condition: action-expert intervention is effective for action-expert-domain failures but not for vision-encoder-upstream perturbations (image blur, viewpoint shift) whose damage propagates from the SigLIP vision encoder before the action-expert layers are reached. This identifies where activation-space shielding works and precisely what architectural extension is needed next.

Conclusion SHIELD establishes that frozen VLA activations contain actionable failure and recovery signal, and that activation-space residuals can selectively rescue action-expert-domain failures without touching the base policy. The central finding is not a limitation but a research result: action-expert intervention works precisely where the forward-pass causal structure permits it, and the boundary it reveals directly motivates the next architectural step—pairing action-expert shielding with earlier visual-encoder intervention for vision-encoder-upstream failures. The right next questions are: (1) can conceptor-based PCA injection into SigLIP's output-potent subspace recover the vision-encoder-upstream failure cases the action-expert cannot reach, extending the same shielding recipe one stage earlier in the forward pass; (2) can a failure-domain classifier gate replace the single global threshold, conditioning intervention on whether the active failure mode is action-expert-domain or vision-encoder-upstream; and (3) do these findings transfer beyond SmolVLA to other VLA backbones with similar vision-encoder / action-expert factorizations.

SHIELD: Failure-Aware Policy Shielding for Frozen Vision-Language-Action Policies

Daniel Contreras-Esquivel
Stanford University

Tianhui (Alina) Huang
Stanford University

Jacob Lee
Stanford University

Abstract

Vision-language-action policies can perform diverse manipulation tasks, but they remain brittle under distribution shift. We study whether a frozen VLA policy can be made safer using an inference-time activation shield. SHIELD / FAPS reads internal SmolVLA activations, predicts near-term failure risk, and conditionally injects a learned low-rank residual into an action-expert transformer layer. The system combines a multi-objective risk head, calibrated hysteresis gating, and a flow-matching residual model constrained to a PCA subspace. Across LIBERO perturbation rollouts, we find that failure risk is predictable from frozen activations via linear probing (wAUC ≈ 0.92); that adding NT-Xent contrastive supervision improves the risk embedding geometry (silhouette score $-0.040 \rightarrow +0.168$) without which the downstream flow model cannot be meaningfully conditioned; and that an NT-Xent-only detector collapses to 0.440 wAUC, confirming supervised risk labels are necessary. Conceptor-guided flow models learn nontrivial activation residuals ($\|\Delta\| \approx 2.0$) compared to near-zero Gaussian and own-encoder controls. Closed-loop evaluation shows selective rescue under action noise, with per-task success rate gains of up to $+0.50$, while vision-encoder-upstream perturbations (image blur, viewpoint shift) do not benefit. We conclude that activation-space shielding at the action-expert level is effective for action-expert-domain failures; extending coverage to vision-encoder-upstream failures requires intervention earlier in the forward pass, at the visual encoder. Code is available at `https://github.com/equalsdaniel/CS224R_robotics_project`.

1 Introduction

Large VLA policies such as SmolVLA map observations and language instructions directly to robot actions. Their appeal is clear: they reuse pretrained representations, support many tasks, and avoid task-specific policy engineering. However, deployment is not only about nominal success. Under perturbations such as camera blur, action noise, viewpoint shifts, or joint noise, a policy can begin drifting toward failure long before the final failed action occurs.

This project asks:

> Can we detect and correct impending manipulation failures from the action-expert layers of a frozen VLA, using a calibrated trigger and a learned activation-space residual, without touching the vision encoder or language backbone?

We intentionally avoid fine-tuning any part of the base VLA. Instead, we wrap the frozen policy with a modular shield that reads and writes only the action-expert transformer layers — the component responsible for fusing visual and language representations into motor commands. The vision encoder and language backbone remain completely untouched. If the shield is inactive, the robot executes exactly the base policy. If the shield detects sufficiently high risk in the action-expert activations,

it injects a low-rank correction into a single action-expert layer. This design is attractive when the base model is expensive to retrain, when safety logic should be auditable, and when it is important to understand precisely which component of the VLA is being intervened upon.

We find that the answer to the research question is yes for action-expert-domain failures and no for vision-encoder-upstream ones, and that this boundary maps precisely onto the causal structure of SmolVLA's forward pass: action-expert layers can be steered to recover from perturbations that enter within the action expert's input space, but cannot undo damage that propagates from the vision encoder upstream. Our contributions are:

- A failure-domain boundary finding: action-expert intervention reliably helps for action-expert-domain failures (action noise, joint noise) but not for vision-encoder-upstream ones (image blur, viewpoint shift), because upstream damage propagates through SigLIP before the action-expert layers are reached. This maps the boundary of activation-space shielding onto a concrete architectural property of SmolVLA, filling a gap left by prior work that applies shielding without distinguishing causal entry points.
- A supervision structure finding: binary risk labels and contrastive geometry serve different, non-redundant roles — BCE is necessary for detection, NT-Xent is necessary for flow conditioning — and a four-config ablation (Runs A–D) shows these roles conflict in AUC but compose correctly end-to-end.
- Evidence that conceptor-guided subspace structure drives nontrivial activation corrections: conceptor-conditioned flow models produce residuals of norm ≈ 2.0, while Gaussian and own-encoder controls collapse to near-zero. This confirms that the risk-conditioned PCA subspace is load-bearing — not a design convenience — and that prior approaches injecting arbitrary residuals would not produce meaningful corrections.
- A system-level diagnosis: risk detection and residual learning both work in isolation, but knowing *when* to apply the residual is the bottleneck for closed-loop success. The weak link is intervention policy, not the detector or the corrector.

2 Related Work

SHIELD sits at the intersection of three lines of work: inference-time activation editing in transformers, policy shielding and runtime failure detection, and structured subspace correction. Together these establish both the feasibility of our approach and the specific gaps it fills.

Vision-language-action policies and robustness. VLA policies combine visual observations, language goals, and action generation in a single model. Benchmarks such as LIBERO [Liu et al., 2023] make it possible to study robustness and transfer across manipulation tasks. Prior work on VLA internals [Gao et al., 2025, Kachaev et al., 2025] shows that early-to-mid transformer layers encode spatial layout and object-grasp binding while late layers encode instruction-action grounding — a functional stratification that motivates both our multi-layer feature fusion design and our choice of injection point. Our work uses a frozen SmolVLA policy [LeRobot Team, 2025] as the base, focusing exclusively on inference-time intervention without any weight updates.

Inference-time activation editing. A line of work in large language models shows that hidden states encode behaviorally meaningful directions that can be shifted at inference without retraining. Representation engineering [Zou et al., 2023] demonstrates that adding learned concept vectors to intermediate residual-stream activations reliably redirects model behavior; inference-time intervention [Li et al., 2023] applies a related idea to steer LLM attention heads toward truthful outputs. The weight-editing literature extends this intuition to persistent changes: ROME [Meng et al., 2022] locates factual associations via rank-1 updates to key-matrix null spaces, and MEMIT [Meng et al., 2023] distributes such edits across layers. These works collectively establish that transformer hidden states are the right substrate for steering — not the action output directly. SHIELD extends this principle from discrete token distributions to continuous robot control, where edits must preserve coherent motor commands across an entire action-expert forward pass.

Policy shielding and runtime failure detection. Traditional shielding wraps a policy with a runtime safety monitor that either halts execution or hands off to a fallback controller [Thananjeyan

et al., 2021, Luo et al., 2023]. SAFE [Gu et al., 2025] is the closest prior work in the VLA setting: it extracts last-layer features, scores failure risk with an MLP or LSTM, and applies conformal prediction to set detection thresholds — but issues a stop signal rather than a correction. SHIELD departs from this paradigm in two ways: it fuses features from functionally stratified layers rather than the last layer alone, and it injects a corrective residual so the frozen policy self-corrects rather than halting.

Structured subspace correction and conceptor-guided steering. Constraining edits to a structured subspace reduces the risk of moving activations off the model's learned manifold. Jaeger's conceptors [Jaeger, 2014] provide a principled tool for this: a conceptor $C = R(R + \alpha^{-2}I)^{-1}$ is a soft projection matrix computed from activation covariance, and boolean algebra over conceptors can express directions that "look like success and not like failure." We use conceptors as training-time subspace teachers to construct flow residual targets. Concurrent work COAST [Shi et al., 2026] independently applies multiplicative conceptor gating to robot policy activations for performance optimization; our approach differs by pairing a learned risk trigger with an additive flow-matching residual for safety-oriented correction rather than always-on steering. Flow matching [Lipman et al., 2023] provides the generative mechanism for learning to move within the subspace, and contrastive supervision [Chen et al., 2020] shapes the risk embedding z_t that conditions the flow.

SHIELD inherits the activation-editing premise from the LLM literature, the corrective-rather-than-halting design from the gap in VLA shielding, and the conceptor subspace machinery from structured steering — combining them into a single inference-time system for frozen robot policies.

3 Method

3.1 Problem Setting: Base Policy and Failure Surfaces

The base policy is SmolVLA, a 500M-parameter VLA with three functional components: a SigLIP vision encoder that processes image observations, a language backbone that encodes task instructions, and an action-expert transformer that fuses both modalities into robot actions. Each component introduces a distinct failure surface — vision-encoder-upstream corruption, instruction drift in long-horizon tasks, and action-expert-domain failures within the action expert's own input space. SHIELD interfaces only in the action-expert layers; the vision encoder and language backbone are never modified.

Inspection of the SmolVLA forward pass reveals a structural property that grounds the action-expert-domain boundary. Proprioceptive state is projected via a linear layer and appended to the token sequence as a suffix, but the VLM attention mask is set so that image and language tokens *cannot attend to state or action tokens* — only the action expert, which processes the full suffix, sees proprioceptive state. This means the action expert's input domain is structurally defined: it receives post-SigLIP visual tokens, post-VLM language tokens, proprioceptive state, and noisy action candidates, while the VLM sees only visual and language tokens. Action-expert-domain failures (action noise, joint noise) corrupt inputs that are exclusive to the action expert by architectural construction; vision-encoder-upstream failures corrupt the representation before it reaches either the VLM or the action expert. The Phase 3 empirical boundary — shielding helps for action noise but not image blur or viewpoint shift — is therefore a direct consequence of this forward-pass structure. Figure 1 shows the full SHIELD pipeline.

3.2 Dataset, Perturbation Taxonomy, and Failure Labels

We evaluate on LIBERO manipulation tasks across three suites (`libero_spatial`, `libero_object`, `libero_long`) using frozen SmolVLA rollouts. The dataset comprises 5,117 episodes spanning nominal conditions and eleven perturbation variants organized into four perturbation families. Each episode carries binary failure labels with horizon $H = 30$ steps, time-to-failure (TTF) targets, and extracted action-expert activations at all six candidate layers. Episode-level splits prevent leakage of adjacent windows from the same trajectory into validation.

Perturbations and failure types are related but do not map one-to-one: perturbations describe the input corruption applied at collection time, while failure types describe the behavioral consequence observed in the robot's actions. Table 1 shows the mapping.

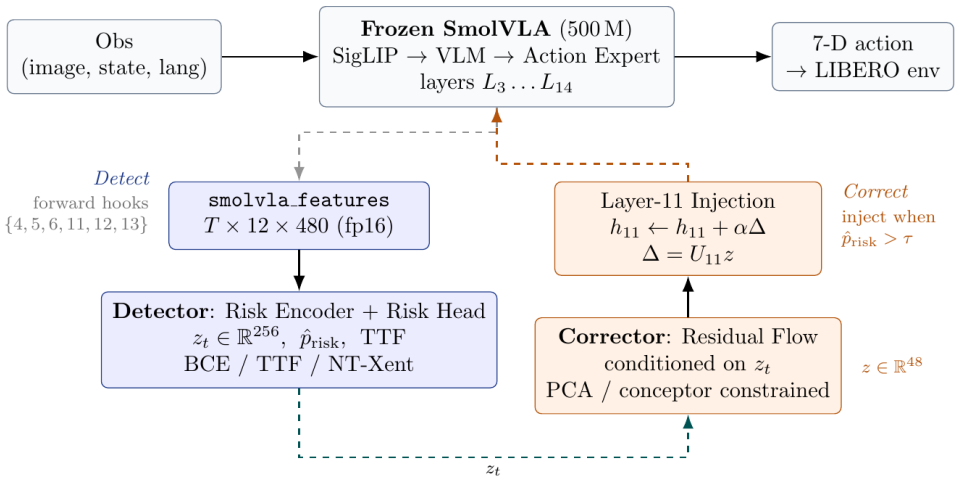

Figure 1: SHIELD wraps a frozen SmolVLA policy with an activation-reading risk head, a calibrated trigger gate, and a low-rank flow residual injector.

Table 1: Perturbation taxonomy and corresponding failure types. Action-expert-domain perturbations corrupt inputs within the action expert's own input space (motor commands, joint dynamics, proprioception); vision-encoder-upstream perturbations corrupt the visual input before the action expert ever sees it. This distinction determines whether action-expert intervention can help.

Perturbation family	Variants	Failure type	Causal stage
Action noise	continuous, intermittent (×3)	`drop`	Action-expert-domain
Joint angle noise	continuous, intermittent	`collision`	Action-expert-domain
Image blur	continuous, intermittent (×3)	`grasp_precision`	Vision-encoder-upstream
Camera viewpoint	continuous	`grasp_misalignment`	Vision-encoder-upstream
None (nominal)	—	`none`	—

Kinematic failure types (`drop`, `collision`) originate in the action-expert layers that SHIELD taps: the visual representation is intact at the time of perturbation. Perceptual failure types (`grasp_precision`, `grasp_misalignment`) originate upstream in the SigLIP encoder: the action-expert receives a corrupted visual representation and SHIELD's injection point is downstream of the damage. This taxonomy is not post-hoc — it is encoded in the data collection pipeline and used as the contrastive supervision signal during risk head training, where NT-Xent positive pairs are defined by shared failure type.

3.3 Activation Features and Layer Selection

During each forward pass, PyTorch hooks capture action-expert hidden states from six candidate transformer layers $\{4, 5, 6, 11, 12, 13\}$. Let h_k denote the hidden activation at layer k. These layers span two functional strata [Gao et al., 2025]: early layers (4–6) encode spatial layout and object-grasp binding; late layers (11–13) encode instruction-action grounding. The injection layer is selected by offline linear probe AUC; layer 11 ranks first (AUC 0.921), with layers 12–13 effectively tied, and is preferred over those alternatives because it sits mid-network, giving the correction more downstream layers to propagate through before the action is decoded. It is used as the single injection point in all Phase 3 evaluations. The base weights are fixed throughout.

3.4 Risk Head

The risk head is a shared encoder with a temporal CNN over a sliding window of activations followed by a cross-layer attention module that fuses the six candidate layers into a single representation. The encoder has 3.76M parameters and outputs a latent embedding $z_t \in \mathbb{R}^{256}$. Three prediction heads branch from z_t:

1. binary horizon risk $y_t = \mathbb{1}[\text{failure within } H \text{ steps}]$,
2. time-to-failure TTF_t, and
3. the contrastive embedding z_t itself, supervised via NT-Xent.

Binary risk detects danger, while TTF distinguishes early-warning states from last-moment failures. Contrastive supervision is useful because the downstream flow model needs a representation with meaningful failure/recovery geometry, not only a high-AUC classifier.

The multi-objective training loss is:

$$\mathcal{L}_{\text{risk}} = w_{\text{bce}}\mathcal{L}_{\text{BCE}} + w_{\text{ttf}}\mathcal{L}_{\text{TTF}} + w_{\text{con}}\mathcal{L}_{\text{NTXent}}.$$

This decomposition is important experimentally: BCE measures whether a state is dangerous, TTF teaches temporal proximity to failure, and NT-Xent shapes the latent space so that failure and recovery states are geometrically useful for the downstream flow model. We ablate the loss weights across Runs A–D to isolate each term's contribution; results are reported in Section 4.

3.5 Calibration and Hysteresis Gate

Raw risk logits are not automatically calibrated probabilities. We apply temperature scaling [Luo et al., 2023] and verify calibration via reliability diagrams and ECE before using risk scores online. The trigger is a hysteresis finite-state machine with a high entry threshold δ_{high}, low exit threshold δ_{low}, and persistence K: the shield enters recovery mode only after $p_t \geq \delta_{\text{high}}$ and exits only after K consecutive steps below δ_{low}. This turns noisy per-step risk predictions into bounded, stable intervention windows.

3.6 Flow-Matching Residual Injection

When the gate is active, a flow model predicts a low-dimensional correction vector in a PCA subspace of layer k's activation space. We compute the basis $U_k \in \mathbb{R}^{d\times 48}$ by retaining the top-48 principal components of training-set activations at layer k, which captures approximately 95% of the activation variance. Conceptors [Jaeger, 2014] provide the training-time residual targets: for each behavioral class (success, recovery, failure), we compute the covariance matrix R of the corresponding activations and derive $C = R(R + \alpha^{-2}I)^{-1}$, a soft projection onto that class's subspace. The flow model is trained to generate corrections that move failure-like states toward the success/recovery conceptor subspace.

At inference, the flow model f_θ maps the current risk embedding z_t and conditioning metadata c to a 48-dimensional correction, which is lifted back to hidden-state space via U_k:

$$z = f_\theta(z_t, c), \qquad \Delta h_k = U_k z, \qquad h'_k = h_k + \alpha \Delta h_k.$$

Constraining edits to this subspace reduces the risk of moving activations far off the SmolVLA manifold. The injection scale α is a tunable parameter at evaluation time and does not require retraining.

3.7 Online Shield Algorithm

At inference time SHIELD executes the following loop:

1. Run frozen SmolVLA on the current observation and language instruction.
2. Read action-expert activations h_k and compute calibrated risk p_t and representation z_t.
3. Enter recovery mode only if $p_t \geq \delta_{\text{high}}$; exit after K consecutive steps below δ_{low}.
4. If recovery mode is active, inject $h'_k = h_k + \alpha U_k f_\theta(z_t, c)$; otherwise leave the hidden state unchanged.
5. Denoise the next action using the frozen policy forward pass.

This separation lets us diagnose failures cleanly: risk detection, residual generation, and intervention timing can each be evaluated independently, which is why each phase of experiments targets a single link in this chain.

Table 2: Experimental phases, research questions, and primary outputs.

Phase	Question	Main output
Phase 1	Can activations predict failure?	Risk head and z_t
Phase 1b	Can risk become a trigger?	Calibration and threshold sweeps
Phase 2	Can we learn residuals?	Flow checkpoints and diagnostics
Phase 3	Does shielding help online?	Closed-loop LIBERO rollouts

Table 3: Risk-head ablation. Run C is selected for downstream flow because it improves latent geometry despite a small AUC cost. AP = average precision.

Run	Losses	wAUC	AP	TTF r	Silhouette
A	BCE	0.928	0.448	0.380	-0.046
B	BCE + TTF	0.927	0.444	0.368	-0.040
C	BCE + TTF + NT-Xent	0.917	0.461	0.413	0.168
D	NT-Xent only	0.440	0.086	0.075	0.125

3.8 Experimental Setup

We evaluate on LIBERO manipulation tasks using frozen SmolVLA rollouts. The dataset comprises 5,117 combined nominal and perturbed episodes, including action noise, image blur, camera viewpoint shift, joint noise, and windowed perturbation variants. Each trajectory carries binary failure labels with horizon $H = 30$, time-to-failure targets, and extracted activation features at all six candidate layers. Risk-head training uses episode-level splits to prevent leakage of adjacent windows from the same trajectory into validation.

Metrics by phase. Each phase answers a distinct question and is evaluated with metrics suited to that question. Phase 1 uses weighted AUC (wAUC) and average precision (AP) to measure risk discriminability, Pearson r against TTF to measure temporal calibration, and silhouette score to measure latent geometry quality. Phase 1b uses ECE and Brier score to measure probability calibration, with temperature T as the calibration parameter. Phase 2 uses flow loss, residual $\|\Delta\|$, and nonzero fraction to measure whether learned corrections are nontrivial relative to the Gaussian and own-encoder controls. Phase 3 uses per-task success rate and false-positive rate (FPR) as the primary closed-loop signals; given small per-condition sample sizes ($n = 10$–50), we report Wilson 95% confidence intervals alongside point estimates. The four phases form a precondition chain rather than independent experiments: a success at each link is necessary but not sufficient for the overall system to work, which is why each phase is reported separately.

Baselines and controls. The primary baseline is unshielded SmolVLA, which establishes the perturbed performance floor without any intervention. Phase 2 includes two diagnostic controls — a Gaussian residual (random injection with matched magnitude) and an own-encoder control (flow model conditioned on its own encoder rather than the risk z_t) — to verify that the learned structure of the correction matters, not merely its presence. Phase 3 compares different trigger and injection settings to isolate where the intervention policy breaks down.

4 Results

4.1 Quantitative Evaluation

With the three-phase pipeline and evaluation protocol established, we now report what each phase found. The quantitative evidence follows the pipeline order: first we test whether risk is readable, then whether risk can be calibrated into a trigger, then whether flow residuals are nontrivial, and finally whether the shield improves closed-loop success.

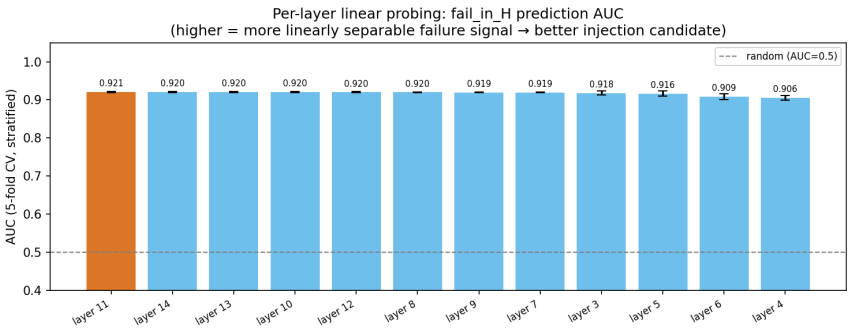

Figure 2: Linear probe AUC across SmolVLA action-expert layers $\{4, 5, 6, 11, 12, 13\}$, averaged over failure types. Layer 11 gives the strongest overall failure/non-failure separability in the probe study (AUC 0.921 ± 0.001) and is selected as the single injection point for all Phase 3 evaluations.

Table 4: Flow residual runs. B6 is the Phase 3 primary checkpoint; C12/C13 and B5 are layer and architecture candidates; D/Z are ablation controls. B6 and B5 learn substantially larger residuals than either control, confirming that the risk-conditioned conceptor subspace drives the learned correction.

Run	Layer	Val. loss	$\|\Delta\|$	Nonzero frac.	Role
flow_b6_real_conceptor_l11	11	0.258	1.999	0.961	**Phase 3 primary**
flow_c4_l12	12	0.300	1.520	0.930	Layer candidate
flow_c5_l13	13	0.303	1.553	0.930	Layer candidate
flow_b5_conceptor_60k	11	0.312	2.000	0.961	Architecture candidate
flow_d_gaussian_l11	11	0.051	0.024	0.758	Gaussian control
flow_z_ownenc_l11	11	0.035	0.082	0.820	Encoder control

4.1.1 Phase 1: Risk is Readable from Frozen Activations

Risk is detectable from frozen activations: BCE-based detectors reach approximately 0.92 weighted AUC (Runs A and B). Run D confirms that contrastive structure alone is insufficient; supervised risk labels are necessary. The TTF Pearson r in Run B (0.368) is slightly lower than Run A (0.380); this is not evidence that TTF hurts temporal calibration, but reflects that jointly optimizing BCE and TTF shifts the latent space geometry in a way the BCE-only probe evaluates less favorably. The improvement becomes apparent in Run C, where NT-Xent further reshapes geometry and TTF r rises to 0.413. Run C is selected for downstream flow conditioning because its silhouette score (0.168 vs. -0.040) reflects substantially better failure/recovery cluster separation, even at a small AUC cost. The retrained Run C checkpoint used in Phase 3 reaches wAUC 0.911 with improved calibration (ECE 0.043).

4.1.2 Phase 1b: Calibration Exposes the Trigger Bottleneck

Initial calibration improved probability quality slightly. For the retrained Run C checkpoint used in final-week evaluation, temperature scaling found $T = 1.5768$, reducing ECE from 0.0704 to 0.0678 and Brier score from 0.0674 to 0.0619. Earlier Run B and Run C calibration sweeps also improved ECE ($0.060 \to 0.052$ for B and $0.061 \to 0.053$ for C), confirming that the detector logits are usable but somewhat overconfident. This matters because thresholds that worked on uncalibrated logits can become too strict after calibration. Indeed, a smoke grid with $\delta_{\text{high}} \in \{0.93, 0.95\}$ produced almost no triggering, while $\delta_{\text{high}} = 0.90$, $\delta_{\text{low}} = 0.55$, and $K = 3$ became the best global selected setting among the tested candidates.

4.1.3 Phase 2: Flow Models Learn Nontrivial Residuals

The Phase 2 result is not simply that the flow loss decreases. The important diagnostic is whether the learned correction has meaningful magnitude and is not just a near-zero target. Conceptor-guided runs (B5, B6) produce much larger residuals ($\|\Delta\| \approx 2.0$) than the Gaussian control (0.024) or own-encoder control (0.082), confirming that the risk-conditioned conceptor subspace drives the

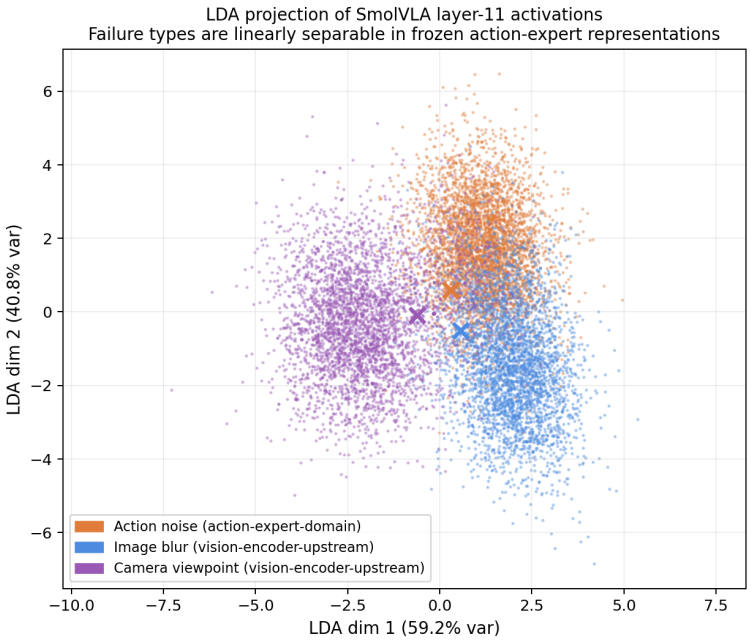

Figure 3: LDA projection of frozen SmolVLA layer-11 activations (480-dim) onto the 2D subspace maximally separating failure types (LDA dim 1: 59.2% variance; dim 2: 40.8%). The three failure types are linearly separable in the raw action-expert representations, confirming that failure-predictive signal exists in the frozen activations before any risk-head training. Action-expert-domain failures (action noise, orange) form a compact, well-separated cluster; vision-encoder-upstream failures (image blur, blue; camera viewpoint, purple) occupy distinct but partially overlapping regions, consistent with their lower eigenvalue mass (Figure 4) and the Phase 3 boundary result.

learned correction rather than arbitrary noise injection. The B6 real-data conceptor checkpoint is selected for Phase 3 because it is trained on full nominal-and-perturbed episodes with per-type failure routing, producing a more broadly conditioned residual than the B5 failure-only run. These offline metrics confirm the components work in isolation but do not predict closed-loop behavioral value: a large residual is useful if the policy is genuinely drifting, but harmful if injected into a nominally successful trajectory. Whether the components compose is the Phase 3 question.

Gate iteration. Phase 3 initially appeared promising because the B6 layer-11 residual could produce qualitative rescues under action noise. The first online runs, however, revealed a different bottleneck: the shield often triggered whenever risk was high, including in rollouts where the frozen policy might have succeeded without intervention. In other words, the failure was not simply "risk is unreadable" or "the residual is useless"; it was that the controller did not yet know when intervention had positive value. We therefore iterated the trigger policy as a sequence of increasingly conservative gates.

Table 5 documents each mechanism and the failure mode it addresses. The aggregate closed-loop results follow.

4.1.4 Phase 3: Online Evaluation is Selective

Closed-loop online evaluation is the hardest test. Early B6 layer-11 demos showed selective rescue on hard action-noise cases, including paired videos where unshielded rollouts timed out and shielded rollouts succeeded. However, after scanning all archived Phase 3 summaries, the aggregate story is more cautious. The early action-noise shielded focus run succeeded in $39/50$ episodes (0.78) but triggered every episode and had mean FPR 0.86. A matched unshielded action-noise baseline over tasks 0–4 reached $43/50$ (0.86). The final-week selected global gate lowered mean FPR to 0.119, but success was $38/50$ (0.76). For all-perturbation tasks 0–2, both strong and medium shield settings reached $71/90$ (0.789), while the unshielded baseline reached $73/90$ (0.811).

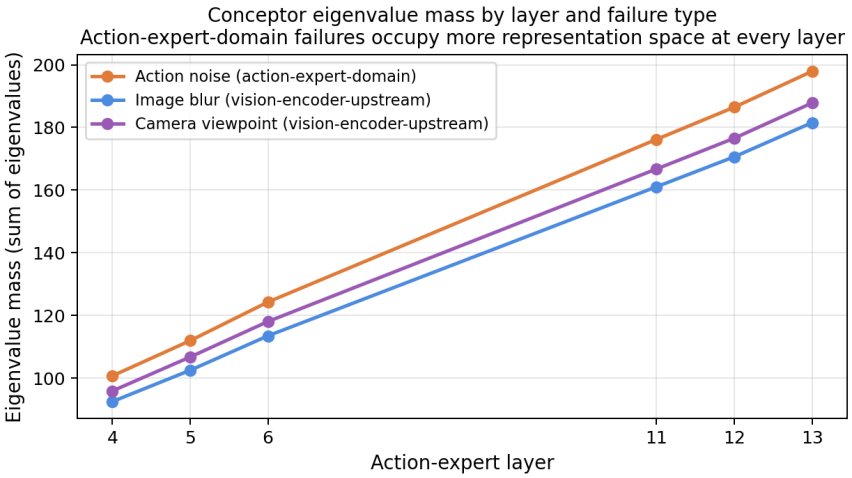

Figure 4: Conceptor eigenvalue mass (sum of eigenvalues of C_f, maximum possible $= d = 480$) by action-expert layer and failure type. Action-expert-domain failures (action noise) consistently occupy more of the action expert's representation space than vision-encoder-upstream failures (image blur, camera viewpoint) at every layer, with the gap widening with depth (layer 4: $+4.8$; layer 13: $+10.1$). This is geometric evidence for the Phase 3 boundary result: at the injection point (layer 11), perceptual failure types have low and flat eigenvalue mass, meaning the action expert forms no structured, high-dimensional subspace for them — so there is no corrective direction for the flow residual to target. Action-noise failures, whose damage enters within the action expert's own input domain (after SigLIP has already produced its visual tokens), accumulate progressively richer representation structure with depth, which is why layer-11 injection can produce meaningful residuals for them. Whether this mass gap reflects SmolVLA's architectural factorization, its training distribution, or both is addressed by the SigLIP probe direction in §6.

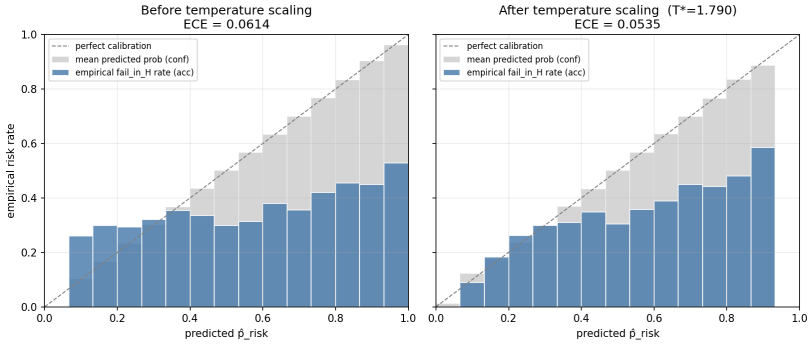

Figure 5: Reliability diagram for the retrained Run C risk head before and after temperature scaling. Calibration makes risk scores more usable as trigger probabilities, reducing ECE from 0.0704 to 0.0678 and Brier score from 0.0674 to 0.0619.

This is not a failure of the entire idea; it is a useful diagnosis. The residual can help in individual recoverable cases, but a single global gate is too blunt for tasks with different perturbation regimes. Wilson 95% confidence intervals for the main aggregates are wide (early B6: $[0.65, 0.87]$; matched baseline: $[0.74, 0.93]$; final gate: $[0.63, 0.86]$), so the stronger conclusion is mechanistic rather than statistical: the calibrated gate reduces false positives relative to early demos, but trigger precision conditioned on perturbation type remains the bottleneck.

The per-task breakdown explains the aggregate result: calibration and hysteresis reduce the worst over-triggering behavior, but the selected global gate still intervenes on tasks where the frozen policy already succeeds at high rates.

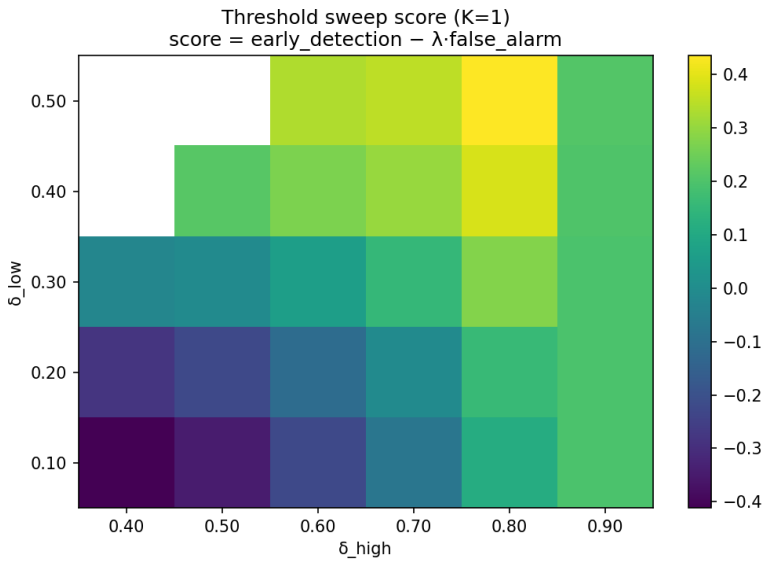

Figure 6: Threshold sweeps expose the central online-control tradeoff: entering recovery early enough to rescue failures while keeping false-positive intervention low on states the frozen policy can still solve.

Table 5: Phase 3 controller iterations. Each mechanism was added to address a concrete failure mode observed in earlier online runs: false alarms on nominal states, threshold chatter, overlong recovery, or intervention when the residual was unlikely to change the action usefully.

Mechanism	What changed	What it diagnoses or prevents
Nominal retraining	Add nominal-success examples to Phase 1 risk-head training	Reduces false alarms on clean states that were never seen during windowed-only training.
Temperature calibration	Scale risk logits before online thresholding	Makes a score such as 0.9 behave more like a probability rather than an arbitrary detector logit.
Hysteresis	Enter at δ_{high}, exit at δ_{low}	Prevents rapid on/off switching when risk hovers near one threshold.
Trigger persistence	Require repeated high-risk steps before entering recovery	Blocks single-step risk spikes from starting intervention.
Recovery budget	Limit the maximum number of recovery steps	Prevents long injection windows that can cause timeouts or disrupt a recovered policy.
Cooldown	Wait after exit before allowing another trigger	Prevents an immediate re-entry loop after the shield has just released control.
Temporal action gate	Trigger only when base actions are changing enough	Avoids intervening when risk is high but the base policy action is stable.
Candidate action gate	Trigger only when the injected candidate action differs enough from the base action	Tests whether the residual would actually change behavior before paying the cost of intervention.
Task-conditioned threshold	Use different thresholds for different task families	Recognizes that easy and hard tasks need different intervention aggressiveness.
Adaptive stall exit	Stop recovery if risk does not decrease	Releases control when the residual is not reducing risk, instead of continuing a failing intervention.

4.2 Qualitative Analysis

The qualitative videos are not used to compute success rates; they explain the behavior behind the numbers in Table 6. They are therefore best read as behavioral diagnostics rather than as additional statistical evidence. In successful action-noise examples, the unshielded policy continues executing plausible but unproductive motions until timeout, while the shielded B6 layer-11 residual changes the action sequence enough to return the robot to a placement trajectory. This supports the Phase 2 conclusion that the flow residual is not merely a low-loss offline artifact: when applied at the right time, it can have a visible corrective effect on closed-loop behavior.

The failure examples show the complementary limitation. When the gate triggers too early, too often, or for too many recovery steps, the same residual can keep the robot moving without completing the task. In these cases the correction is not obviously meaningless; it is mistimed or applied in a

Table 6: Phase 3 evidence synthesis from archived run summaries. The learned residual can produce successful recoveries, but aggregate comparisons show that global triggering is not yet reliable enough. The population column is included because the action-noise rows and all-perturbation rows are not directly comparable benchmarks.

Run family	Population	Episodes	SR	Trigger	FPR
Early B6 action-noise focus	Tasks 0–4, action noise	50	0.78	1.00	0.860
Matched action-noise baseline	Tasks 0–4, action noise	50	0.86	0.00	0.000
Final selected global gate	Tasks 0–4, action noise	50	0.76	0.84	0.119
Strong all-perturbation gate	Tasks 0–2, all perturb.	90	0.789	0.956	0.567
Medium all-perturbation gate	Tasks 0–2, all perturb.	90	0.789	1.00	0.685
Unshielded all-perturbation baseline	Tasks 0–2, all perturb.	90	0.811	0.00	0.000

Table 7: Final-week selected global threshold on action-noise tasks, with matched unshielded SmolVLA baselines. ΔSR is Shield SR minus Base SR. The shield improves tasks 0–1 but hurts tasks 2–4, showing why a single global trigger is too blunt even after calibration.

Task	Shield SR	Base SR	ΔSR	Trigger	FPR	TPR	Precision
0	0.70	0.60	+0.10	0.80	0.189	0.095	0.192
1	1.00	0.90	+0.10	0.90	0.036	0.057	0.855
2	0.70	1.00	-0.30	0.80	0.223	0.018	0.036
3	0.60	0.80	-0.20	0.80	0.107	0.072	0.155
4	0.80	1.00	-0.20	0.90	0.040	0.032	0.378
Mean	0.76	0.86	-0.10	0.84	0.119	0.055	0.323

state where the base policy may already have been recoverable. This is why the qualitative evidence matches the quantitative result rather than contradicting it: individual rescue clips demonstrate recoverable signal, while the matched-baseline tables show that the current global gate does not yet select those interventions reliably.

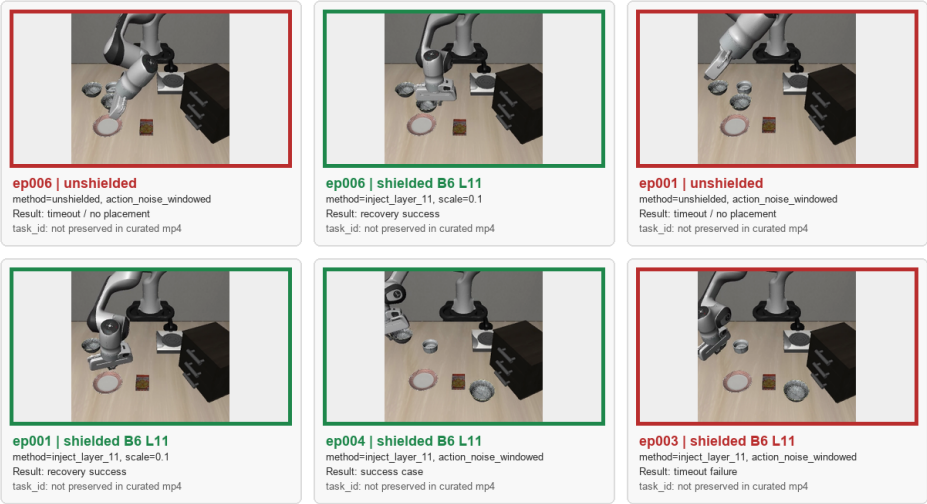

Figure 7: Qualitative Phase 3 action-noise outcomes. Red-bordered panels show timeout failures and green-bordered panels show successful shielded B6 layer-11 recoveries. The gallery illustrates the selective-rescue diagnosis: residual injection can visibly redirect individual action-noise rollouts, but an imperfect gate can still over-apply the same correction and produce timeout failures.

5 Discussion

The phase-by-phase results above confirm that each component works in isolation. The question is why they do not compose into a uniformly stronger closed-loop system — and what that tells us about the boundary conditions of activation-space shielding. The archived runs separate the SHIELD

Table 8: Compact matched-baseline checks on completed Phase 3 subsets. These rows are reported separately because they answer different questions: the first is the final selected global-gate action-noise subset, while the second is a targeted hard-task diagnostic for the action/candidate gate. They should not be read as one unified benchmark.

Setting	Subset	Shield / Base SR	Trigger / FPR	Takeaway
Global selected gate	tasks 0–4; $n = 50$	0.760 / 0.860	0.840 / 0.119	Reduces FPR versus eager triggering, but trails matched unshielded.
Action-gate hard-task confirmation	tasks 2–4; $n = 30$	0.833 / 0.933	0.733 / 0.123	Lower FPR than eager settings, but still below matched unshielded.

system into three causal links. The first, *risk readability*, works: Phase 1 reaches high AUC and the calibrated model produces bounded, usable risk scores from frozen SmolVLA activations. The second, *residual learnability*, also works in an offline sense: conceptor-guided B/C-family flow models learn nontrivial activation corrections, while the Gaussian and own-encoder controls collapse toward near-zero residuals, confirming that the learned structure matters. The third link, *intervention policy*, is the weak link. Online success depends on the product of detection quality, correction strength, and trigger timing:

$$\mathbb{E}[\Delta\mathrm{SR}] \approx P(\mathrm{trigger} \mid \mathrm{recoverable}) \cdot G_{\mathrm{rescue}} - P(\mathrm{trigger} \mid \mathrm{safe}) \cdot C_{\mathrm{disruption}}.$$

The current global gate reduces $P(\mathrm{trigger} \mid \mathrm{safe})$ relative to early demos, but does not condition on which perturbation type is active or whether the corruption entered the forward pass before the action expert. Under action noise, the perturbation is action-expert-domain: SmolVLA's visual representation is intact and the action-expert layers contain the failure signal. Here G_{rescue} is positive, and the same correction that harms a nominally easy rollout can rescue a perturbed one. Under image blur or camera viewpoint shift, the perturbation is vision-encoder-upstream: it corrupts the visual input before the action expert forms its representation, so corrections injected at layers $\{4, 5, 6, 11, 12, 13\}$ arrive downstream of the damage and G_{rescue} is near zero. This explains why qualitative paired rescues and negative aggregate results can both be true simultaneously—they arise from the same mechanism applied to perturbations with different causal entry points in the forward pass.

This failure-domain boundary is not merely an empirical observation; inspection of the SmolVLA forward pass (§3.1) reveals it is structurally enforced. Proprioceptive state is exclusive to the action expert by the VLM attention mask, so action-expert-domain failures corrupt inputs the action expert owns. Vision-encoder-upstream failures corrupt the representation before it reaches either the VLM or the action expert, so no intervention at action-expert layers can recover the lost information. This is not a general limit of activation steering; it is a precise limit determined by where in the forward pass the damage enters. A two-stage architecture — action-expert shielding for action-expert-domain failures, visual-encoder shielding for vision-encoder-upstream failures — could address both failure modes without touching the base policy weights.

6 Future Directions

The boundary finding in §5 is the starting point for a longer research program, not a stopping point. The natural arc is a modular multimodal shield where each VLA component — action expert, visual encoder, language backbone — has its own failure probe, subspace basis, and flow corrector, all sharing the same gating and calibration infrastructure developed here. Getting there is not straightforward: each new encoder brings its own activation geometry, failure taxonomy, and subspace structure, and transfer between architectures will require re-probing and re-fitting rather than drop-in reuse. The gate problem also has to be solved per-deployment — a learned intervention-value model trained on LIBERO won't generalize to a different robot or task distribution without retraining on matched rollouts. What does transfer is the methodology: the probe-then-correct recipe, the conceptor subspace machinery, and the three-phase evaluation structure. The data infrastructure for the next round of experiments is already in place: the 5,117-episode archive, pre-computed U_{potent}/U_{null} conceptors, and calibrated risk embeddings support all near-term experiments without new rollout collection.

What we want to do next. All three directions below require no new rollouts and no architectural changes — the data and subspace machinery are already in place.

- **Fix the gate.** The Phase 3 bottleneck is trigger policy, not detection or correction. A small learned gate conditioned on calibrated risk, risk slope, failure-type posterior, and candidate-action disagreement replaces the global threshold with a decision that actually conditions on whether intervention helps. The $U_{\text{potent}}/U_{\text{null}}$ subspace bases are already computed; an A/B injection comparison at layer 14 (zero downstream wash-out) vs. layer 11 is the first ablation to run once the gate is in place.
- **Extend shielding to the visual encoder and language backbone.** The SigLIP probe is the most obvious next step: if the conceptor eigenvalue mass pattern inverts at SigLIP layers — image blur showing *higher* mass there than action noise — that confirms the action-expert mass gap is architectural, not a training artifact, and grounds a principled injection point for perceptual failures. SigLIP also has language baked in from contrastive pretraining, so a single probe pass may expose both visual and semantic failure signal. For long-horizon tasks, hooking into `vlm.model.text_model.layers` on existing `libero_long` episodes would show at which depth instruction drift becomes detectable — no new rollouts needed.
- **Learn a value-of-intervention policy end-to-end.** The longer version of this project freezes the VLA entirely and fine-tunes only the flow model with RL in LIBERO simulation, using task success as the reward. The current FM checkpoint is a strong warm start; RL would discover corrections that actually maximize recovery rate rather than match a synthetic target direction. The probability-flow ODE is differentiable, so policy gradients flow through $x_1 = x_0 + \int v_\theta \, dt$ back to θ without a score-function estimator.
- **Real-hardware evaluation.** LIBERO simulation removes unmodeled dynamics, sensor noise, and actuation variability that would stress the shield in ways the current eval cannot measure. A real-robot eval on a Franka or similar platform — collecting perturbed rollouts, fitting SHIELD on that data, and testing closed-loop recovery — would determine whether the failure geometry and subspace structure found in simulation survive contact with physical hardware. This is the natural next collaboration step: access to real manipulation hardware and the expertise to run rigorous robot learning experiments would let us test whether the probe-then-correct recipe transfers out of simulation, and what re-fitting is required when it does not.

7 Conclusion

SHIELD shows that frozen VLA activations contain real failure signal and that learned activation-space residuals can rescue action-expert-domain failures without touching the base policy. The clearest result is a boundary: shielding works where the perturbation enters within the action expert's input space, and doesn't where the damage is already baked into the visual representation before the action expert sees it. That boundary isn't a limitation so much as a map — action-expert shielding for kinematic failures, visual-encoder intervention for perceptual ones, language-backbone probing for instruction drift in long-horizon tasks. Each modality needs its own probe and steering mechanism, but the machinery developed here transfers directly. There's a clear next experiment for each.

8 Team Contributions

- **Daniel Contreras-Esquivel:** proposed the project concept and the central research question; contributed to the three-phase pipeline design and the conceptor-based subspace framework; built much of the data and evaluation infrastructure (rollout collection, failure labeling, layer activation extraction, PCA/null/potent basis construction, per-type failure conceptors, and closed-loop Phase 3 evaluation); contributed visualization and diagnostic tooling; and helped with poster design.
- **Tianhui (Alina) Huang:** led Phase 1 risk-head experiment design and training, including Runs A–D, calibration, threshold sweeps, Run C representation selection, and nominal-success retraining; contributed to Phase 2/3 system and ablation design around layer-11 PCA injection, conceptor-guided flow variants, B5/B6 checkpoint selection, D/Z controls, and final-week trigger/gate redesign; co-authored system design and evaluation planning;

led future-work value-gated architecture design and the task plan for detector scaling, residual routing, paired rollout collection, and intervention-value gating; led W&B/Modal run management, result synthesis, ablation planning, public-release materials, and proposal, milestone, poster, final-report, and website writing; implemented key engineering improvements for training throughput, logging reliability, Phase 3 evaluation speed, and trigger/gate correctness.

- **Jacob Lee:** executed the windowed-perturbation episode collection pipeline on Stanford Farmshare — collecting the recovery supervision dataset (action noise, image blur, camera viewpoint across cutoff steps 50/100/150) that Phase 2 flow training depends on for positive recovery examples; ran Phase 3 online evaluation jobs and archived results; contributed paired shielded/unshielded comparison videos demonstrating qualitative rescue behavior; and participated in Phase 2 flow and conceptor experiment runs and interpretation of residual magnitude and shield behavior across conditions.

Acknowledgments. The authors thank Ke Wang, Marcel Torné, and Perry Dong for TA mentorship, and Chelsea Finn for faculty guidance throughout the project. We also thank Modal Labs for generously providing compute credits that supported data collection and model training.

Changes from Proposal. The proposal focused on failure-aware activation steering and recovery, with IQL and flow matching as parallel residual learners. During implementation the IQL branch was deferred in favor of deeper investigation of the flow-matching and conceptor-guided residual approach. The pipeline was refined into three explicit phases: risk detection and calibration, flow residual learning, and closed-loop online evaluation. The biggest change from the proposal is that the final conclusion is more conservative than the original ambition: risk detection and residual learning both work, but global triggering is the dominant bottleneck for closed-loop success.

Related concurrent work. A parallel CS 231N project by D. Contreras-Esquivel, "Probing SigLIP's Spatial Structure: Object and Scene Factorizability in SmolVLA," independently probes the same frozen vision encoder from a different angle: whether SigLIP spontaneously encodes factorizable object and scene manifolds across its transformer layers, and whether the entanglement between those manifolds is baked in by contrastive pretraining or introduced by global self-attention at inference time.[1] The key findings — a sharp intrinsic-dimensionality collapse at layer 5 where object representations compress from $\sim$100 PCA components to 2–3 while scene dimensionality expands, structured implicit feature binding at that same layer, and a derived object-exclusive projection matrix $P_{\text{obj}} \in \mathbb{R}^{768\times 32}$ via residual subspace decomposition — directly motivate the SigLIP probe direction in Future Directions: if the eigenvalue mass pattern inverts at SigLIP layer 5, it would ground a principled injection point for vision-encoder-upstream failures. The two projects share the rollout dataset and SmolVLA infrastructure but have non-overlapping hypotheses and methods.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, and Lin Shao. VLA-OS: Structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv preprint arXiv:2506.17561*, 2025.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask failure detection for vision-language-action models. In *Advances in Neural Information Processing Systems*, 2025.

Herbert Jaeger. Controlling recurrent neural networks by conceptors, 2014.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K. Kovalev, and Aleksandr I. Panov. Don't blind your VLA: Aligning visual representations for OOD generalization. *arXiv preprint arXiv:2510.25616*, 2025.

[1] Unpublished course project; code at `https://github.com/equalsdaniel/cs231nproject`.

LeRobot Team. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. Project documentation and model release.

Kenneth Li, Oam Patel, Fernanda Viégas, Hans Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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*, 2023.

Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. *International Journal of Robotics Research*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

Yixuan Shi et al. COAST: Conceptor-based activation steering for vision-language-action models, 2026. Concurrent work.

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. In *IEEE Robotics and Automation Letters / ICRA*, 2021.

Andy Zou, Long Phan, Sarah Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J. Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2023.