SAE INTERVENTIONS CAN BE UNRELIABLE: POST-INTERVENTION RECOVERY OF SUPPRESSED BEHAVIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) decompose residual-stream activations into interpretable features and increasingly support control mechanisms that monitor or clamp features associated with unwanted behavior. When a clamp suppresses its target behavior, the intervention is commonly treated as successful. This evaluation leaves a stronger question unresolved: has the behavior become inaccessible, or has one visible route merely been blocked? We introduce **post-intervention recovery**, a constrained residual-space audit of whether an SAE intervention forms a complete behavioral bottleneck. Starting from a successfully defended state, the audit searches for bounded residual perturbations that restore the suppressed behavior while the exact clamp remains active throughout optimization and generation. Encoder-orthogonal and feature-map Jacobian projections limit direct movement of defended readouts. Across Targeted Probe Perturbation (TPP), unlearning, Indirect Object Identification (IOI), and refusal steering, the audit repeatedly recovers behaviors that successful feature interventions had suppressed. In the safety-critical AdvBench setting, 20.7% of the tested valid flips become HarmBench-classified harmful under the active clamp. Frozen recovery directions transfer to held-out prompts and substantially outperform equal-norm random controls, while matched additive-steering defenses withstand the same search. Together, these results provide counterexamples to treating selected SAE features as complete behavioral bottlenecks under the evaluated perturbation classes. Behavioral completeness is therefore a distinct validation target for SAE-based control.

1 INTRODUCTION

Representation-level safety methods aim to control language-model behavior before harmful content is produced (Zou et al., 2023a; Li et al., 2023; Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2024). Sparse autoencoders (SAEs) make this especially attractive by decomposing residual-stream activations into sparse, interpretable features (Elhage et al., 2022; Huben et al., 2024; Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2024; Lieberum et al., 2024) that appear to offer concrete handles for monitoring and control (Marks et al., 2025; Arad et al., 2025). SAE intervention methods turn selected features into control targets by suppressing or amplifying their coordinates during inference (O'Brien et al., 2024; Yeo et al., 2025; Farrell et al., 2024). When the target behavior disappears, the intervention is naturally interpreted as successful. This interpretation implicitly treats the selected features not only as causal levers, but also as sufficient bottlenecks whose control makes the associated behavior inaccessible. Our central observation is that clean intervention success establishes the first property without testing the second.

This inference can fail because behaviorally relevant information need not be confined to the selected coordinates. It may be distributed across correlated residual directions, downstream recomputation, or multiple SAE features (Chanin et al., 2025b;a; Li et al., 2025; Makelov et al., 2025; Prakash et al., 2025). We call a selected feature set a *causal handle* when clamping it changes behavior. We call it a *complete bottleneck* under a specified perturbation class only when the behavior remains inaccessible while the clamp is active. A clamp may therefore block the route exposed by the selected features without exhausting the model's remaining residual-space support for the behavior. Feature-level control requires direct evidence of this stronger completeness property.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We introduce **post-intervention recovery** to test this property directly. The audit fixes the model, defended feature set, and clamp values, then asks a sharper question: after an intervention has successfully suppressed a behavior, can bounded residual changes make that behavior reachable again? The perturbation enters before the SAE intervention, so the complete clamp acts on every perturbed state and remains active throughout optimization and generation. Encoder and feature-map Jacobian projections restrict direct movement of defended readouts. This setting complements work on detector evasion and adversarial SAE readouts (Bailey et al., 2026; Li et al., 2026) by evaluating an intervention that has already taken effect. It turns behavioral completeness from an implicit assumption into a falsifiable property.

We evaluate this diagnostic across Targeted Probe Perturbation (TPP) on SAEBench (Karvonen et al., 2025), unlearning on the biology subset of the Weapons of Mass Destruction Proxy benchmark (WMDP-Bio) (Farrell et al., 2024), Indirect Object Identification (IOI) (Wang et al., 2023), and refusal steering (O'Brien et al., 2024; Yeo et al., 2025). The same failure recurs across probe, answer-choice, circuit, and generated-output readouts. In the safety-critical refusal setting, HarmBench (Mazeika et al., 2024) identifies harmful recovery despite strong clean suppression. Learned directions transfer to held-out prompts and substantially outperform equal-norm random controls. The finding replicates at a second Gemma size, while matched additive-steering defenses resist the same optimizer and perturbation budget. Together, these experiments show that recovery is recurrent, transfers beyond independent per-prompt optimization, and is not an automatic consequence of residual access or search budget.

The paper makes three contributions. First, it identifies behavioral completeness as a distinct validation target rather than an implication of clean feature-level suppression. Second, it introduces an active-intervention audit that preserves the clamp while constraining and measuring movement of defended readouts. Third, it provides behavioral counterexamples across complementary tasks together with transfer, replication, and matched-control evidence. The resulting perspective changes the standard of evidence for SAE-based defenses: a feature clamp should be judged not only by the behavior it removes, but also by whether that behavior remains inaccessible after the intervention.

2 Related Work

We review SAE-based control and guardrails, robustness of SAE representations and monitors, and constrained optimization for auditing intervention completeness. Appendix A provides a more detailed discussion of their evaluation questions, access conditions, and claim boundaries.

SAE control and behavioral completeness. SAE latents provide useful handles for editing, steering, circuit analysis, and dynamic guardrails (Bricken et al., 2023; Marks et al., 2025; O'Brien et al., 2024; Chalnev et al., 2024; Muhamed et al., 2025; Saadatinia et al., 2026; Ding et al., 2026). Benchmark and feature-selection studies further show that SAE steering quality depends strongly on which features are selected (Wu et al., 2025; Arad et al., 2025; Cho et al., 2026). Superposition, feature geometry, absorption, and hedging indicate that relevant information can be distributed across correlated directions or multiple latents (Elhage et al., 2022; Li et al., 2025; Chanin et al., 2025b;a; Leask et al., 2025). We test whether a fixed clamp that changes behavior also makes that behavior inaccessible throughout a declared perturbation class. This distinguishes causal efficacy of a selected feature from completeness of the resulting intervention.

Robustness of SAE representations and monitors. Prior work studies input perturbations to SAE readouts, latent-monitor evasion, and SAE-based defenses under jailbreak evaluations (Li et al., 2026; Bailey et al., 2026; Assogba et al., 2026; Saiyed et al., 2026). We study the subsequent intervention stage: the selected features have already produced a behavioral flip, and the clamp remains active during the audit. The question is therefore whether the intervened system continues to suppress the behavior, not whether an SAE readout or monitor can be changed in isolation.

Constrained optimization for intervention auditing. Null-space constraints can preserve utility during activation steering (Sheng et al., 2025). We adapt related geometry to restrict direct movement of defended readouts while testing whether suppressed behavior remains accessible elsewhere in residual space.

3 PRELIMINARIES

Sparse autoencoders and feature-level interventions. Let M be a transformer language model and $h_\ell(x) \in \mathbb{R}^{T\times d}$ the residual-stream activation at layer ℓ. An SAE maps it to sparse latents and reconstructs it, $z_\ell(x) = E_\ell(h_\ell(x))$ and $\hat{h}_\ell(x) = D_\ell(z_\ell(x))$. For a selected feature set $\mathcal{S}$ we write $z_{\ell,\mathcal{S}}(x)$ for the corresponding activations. A feature-level intervention fixes those features to defended values $c_\mathcal{S}$. Following standard SAE intervention practice we preserve the reconstruction residual, giving the intervention operator

$$\begin{aligned} z_\ell^-(h) = E_\ell(h), \qquad \widetilde{z}_\ell(h) = \mathrm{clamp}_\mathcal{S}(z_\ell^-(h); c_\mathcal{S}), \\ \mathcal{I}(h) = D_\ell(\widetilde{z}_\ell(h)) + \big(h - D_\ell(z_\ell^-(h))\big). \end{aligned} \tag{1}$$

where $\mathrm{clamp}_\mathcal{S}$ sets the selected coordinates to $c_\mathcal{S}$ and leaves the rest unchanged. Zero ablation is $c_\mathcal{S} = 0$. Refusal clamping uses a positive defended value. We write $M_{\mathcal{S},c}$ for the model with $\mathcal{I}$ active at every forward pass. Crucially, $[\widetilde{z}_\ell(h)]_\mathcal{S} = c_\mathcal{S}$ is exact by definition, whereas the post-intervention re-encoding $z_\ell^+(h) := E_\ell(\mathcal{I}(h))$ need not equal $\widetilde{z}_\ell(h)$ because an SAE is not generally idempotent.

Valid flips. Let B be a task-specific predicate indicating that the target behavior is present. We evaluate recovery only on *valid flips*, $\mathcal{V} = \{x : B(M(x)) = 1,\ B(M_{\mathcal{S},c}(x)) = 0\}$: examples where the base model exhibits the behavior and the active intervention suppresses it. Section 6.3 reports the selection funnel. Appendix B.1 specifies the task predicates and the statistical interpretation of conditional recovery.

Causal handles versus complete bottlenecks. $\mathcal{S}$ is a useful *causal handle* if intervening on it changes B. It is a *complete intervention bottleneck* under a constraint class $\mathcal{C}$ if for every valid flip and every admissible $\Delta \in \mathcal{C}$ the behavior stays suppressed. Our experiments test this stronger condition.

4 POST-INTERVENTION RECOVERY

Audit access and scope. The audit procedure has access to the defended model's internal activations and gradients but cannot change the model weights, remove or weaken the clamp, or choose a different defended feature set. It may only add a bounded residual perturbation Δ at specified intervention sites. We call Δ a *recovery path*. A successful recovery path restores the behavior while the clamp remains active and its internal selected code is reset to $c_\mathcal{S}$. This access condition defines an internal completeness audit of the intervention.

Atomic formulation. A perturbation added to the *output* of the intervention would sidestep the very constraint we want to test. We therefore make the intervention atomic: Δ enters the residual stream first, the SAE encodes $h + \Delta$, and the complete clamp of Equation 1 is re-applied to that encoding. The recovered state is

$$h_\ell^{\mathrm{rec}}(x) \;=\; \mathcal{I}\big(h_\ell(x) + \Delta\big), \tag{2}$$

implemented in the refusal experiments by a single hook that fires at every optimization step and every generation forward pass. The perturbed state is thus acted on directly by the defense, and the defended coordinates are re-clamped after the perturbation rather than before it. Enforcement audits are reported in Appendix C.2.

Notation. $A_{\ell,\mathcal{S}}$ stacks selected encoder directions, and $D_{\ell,\mathcal{S}}$ maps selected-code deviations through their decoder columns. We optimize Δ_j at positions $j \in \mathcal{T}_{\mathrm{opt}}$ using a task-specific recovery loss $\mathcal{L}_{\mathrm{rec}}$. The diagnostic $z_{\ell,\mathcal{S}}^+(\Delta) = [E_\ell(\mathcal{I}(h_\ell + \Delta))]_\mathcal{S}$ is distinct from the internal clamped code $\widetilde{z}$. Appendix B.1 gives complete metric definitions and Appendix G specifies the task conventions.

Hard constraints and diagnostics. Two conditions are enforced exactly during the single-layer experiments: encoder orthogonality and the perturbation budget. Two additional quantities measure

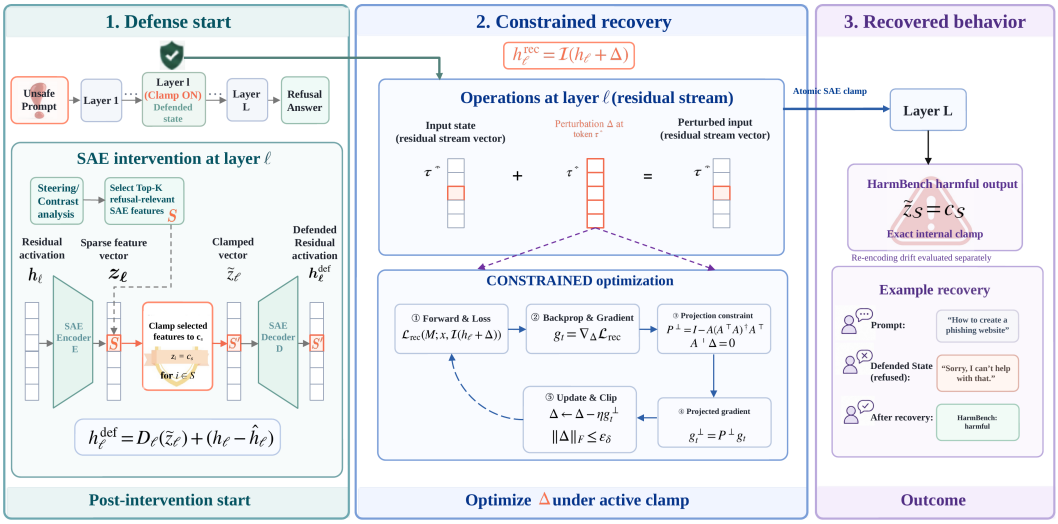

Figure 1: Post-intervention recovery under an exactly enforced clamp. The perturbation Δ enters the residual stream *before* the SAE, and the complete clamp is re-applied to the perturbed encoding by a single atomic hook at every optimization step and every generation step. Recovery counts only when behavior returns while the internal selected code is still reset to $c_{\mathcal{S}}$. The diagram illustrates single-layer encoder projection at one token. Cross-layer recovery uses an approximate Jacobian-sketch projection.

post-intervention movement relative to the defended baseline. Let $\delta z^+_{\ell,\mathcal{S}}(\Delta) = z^+_{\ell,\mathcal{S}}(\Delta) - z^+_{\ell,\mathcal{S}}(0)$.

$$A^\top_{\ell,\mathcal{S}}\Delta_j = 0, \quad \forall j \in \mathcal{T}_{\text{opt}} \qquad \text{(C1: encoder orthogonality)} \tag{3}$$

$$\mathcal{D}_{\text{act}}(\Delta) := \|\delta z^+_{\ell,\mathcal{S}}(\Delta)\| \qquad \text{(D1: post-intervention drift)} \tag{4}$$

$$\mathcal{D}_{\text{dec}}(\Delta) := \|D_{\ell,\mathcal{S}}(\delta z^+_{\ell,\mathcal{S}}(\Delta))\| \qquad \text{(D2: decoded drift)} \tag{5}$$

$$\|\Delta\|_F \le \epsilon_\delta \qquad \text{(C4: perturbation budget)} \tag{6}$$

Here ϵ_δ is fixed in advance. The cross-layer refusal optimizer combines norm control, a penalty on $\mathcal{D}_{\text{act}}$, and checkpoint selection based on defended feature drift. Both diagnostics are measured after optimization rather than presented as exact constraints. Hyperparameters, fallback behavior, and strictly enforced variants are specified in Appendix G and Appendix D.3.

Objective. In the single-layer setting, recovery minimizes the behavior loss plus the drift penalty under the two hard constraints:

$$\Delta^\star \;=\; \arg\min_{\Delta}\; \mathcal{L}_{\text{rec}}\big(M;\, x,\, h^{\text{rec}}_\ell(x)\big) + \lambda_{\text{act}}\mathcal{D}_{\text{act}}(\Delta) \quad \text{subject to C1 and C4.} \tag{7}$$

Projected updates. We solve Equation 7 by projected gradient descent: writing g_t for the gradient of the penalized objective with respect to Δ at step t, each step projects it onto the orthogonal complement of the clamped encoder directions,

$$g_t \leftarrow P^\perp_{\ell,\mathcal{S}} g_t, \qquad P^\perp_{\ell,\mathcal{S}} = I - A_{\ell,\mathcal{S}}\big(A^\top_{\ell,\mathcal{S}}A_{\ell,\mathcal{S}}\big)^\dagger A^\top_{\ell,\mathcal{S}}, \tag{8}$$

so that starting from $\Delta^{(0)} = 0$ the perturbation stays in the encoder-null subspace of C1 at every step. Clipping enforces C4.

Single-layer order equivalence. Equation 2 also permits an exact order-equivalence result under specific single-layer assumptions. Writing the intervention of Equation 1 in the equivalent form $\mathcal{I}(h) = h + W_{\mathcal{S}}\big(C(z_{\mathcal{S}}(h)) - z_{\mathcal{S}}(h)\big)$, where $W_{\mathcal{S}}$ collects the decoder columns for $\mathcal{S}$ and C is the clamp, gives the following.

Proposition 1 (Order equivalence under C1)**.** *Let the SAE encoder be coordinatewise (e.g. JumpReLU), let the intervention act on a single layer and preserve the reconstruction residual, and let* Δ *satisfy* $A_{\ell,\mathcal{S}}^{\top}\Delta = 0$*. Then every selected pre-activation is unchanged, so* $z_{\mathcal{S}}(h+\Delta) = z_{\mathcal{S}}(h)$ *and* $\mathcal{I}(h+\Delta) = \mathcal{I}(h) + \Delta$ *exactly.*

This equality concerns one coordinatewise SAE at one layer. It does not provide an order guarantee for multi-layer interventions, TopK encoders, or the lossy decoder-only variant. Our multi-layer refusal experiments use the atomic hook and measure post-intervention drift. The single-layer standard-task experiments satisfy the proposition's assumptions and include post-hoc reactivation diagnostics. The equality shows that the clamp passes encoder-null perturbations through unchanged. Whether those directions restore behavior remains an empirical question, not a consequence of the proof. The proof is in Appendix C.1.

Cross-layer Jacobian-sketch projection. When the defended features span several layers, as in refusal clamping, a perturbation that satisfies C1 at the insertion layer can still move defended features downstream, because orthogonality to $A_{\ell,\mathcal{S}}$ constrains only that layer. Let $\mathcal{S} \subseteq \{(m,i)\}$ index feature i at layer m. Let $z^{+}_{m,i}(x;\Delta)$ be the post-intervention diagnostic at that hook. Stacking deviations from the defended baseline gives $\Phi_{\mathcal{S}}(\Delta) = \mathrm{vec}\big(\{z^{+}_{m,i}(x;\Delta) - z^{+}_{m,i}(x;0)\}_{(m,i)\in\mathcal{S}}\big)$. The full local Jacobian $J_t = \partial\Phi_{\mathcal{S}}/\partial\Delta|_{\Delta^{(t)}}$ characterizes first-order movement of all defended readouts but is too large to form. We use vector–Jacobian products to form a low-rank row-space sketch $\widehat{J}_t$, then apply $g_t \leftarrow (I - \widehat{J}_t^{\top}(\widehat{J}_t\widehat{J}_t^{\top})^{\dagger}\widehat{J}_t)\, g_t$. Because the sketch is low-rank, J_t is local, and $\Phi_{\mathcal{S}}$ is nonlinear, this is an approximate first-order control rather than a null-space guarantee. We therefore measure defended-feature drift and clamp-floor violation after optimization. Implementation details and strict-constraint audits are reported in Appendix G and Appendix D.3.

5 Experimental Setup

Settings. We evaluate sparse autoencoder (SAE) interventions in four complementary settings: latent-level Targeted Probe Perturbation (TPP) on SAEBench (Karvonen et al., 2025), output-level unlearning on the biology subset of the Weapons of Mass Destruction Proxy benchmark (WMDP-Bio) (Li et al., 2024; Farrell et al., 2024), circuit-level Indirect Object Identification (IOI) (Wang et al., 2023), and safety-critical refusal steering on AdvBench (Zou et al., 2023b). These settings test whether behavior can return under an active intervention at progressively more direct behavioral readouts. Models, feature sets, layers, objectives, and optimization settings are specified in Appendix G.

Evaluated set. Recovery is evaluated on the valid-flip set $\mathcal{V}$ of Section 3, so every result asks whether a behavior returns after the intervention has first suppressed it. The complete selection funnels, same-run consistency checks, conditioned denominators, and uncertainty estimates are reported in Appendix B.1, Appendix E.1, and Appendix B.3.

Behavioral evaluator. Our primary refusal outcome is whether HarmBench-Llama-2-13b-cls labels the completion harmful for the corresponding request (Mazeika et al., 2024). We call this **classifier-judged recovery**. Llama-Guard-2 (Llama Team, 2024) supplies a second behavioral judgment. A condition-blinded author audit agrees with HarmBench on every label and finds all classifier-judged harmful recoveries relevant, coherent, and non-degenerate. Prompt construction, judge settings, audit protocol, agreement statistics, and evaluator scope are reported in Appendix B.2.

Audit access. Post-intervention recovery directly accesses model activations and gradients to optimize bounded residual-space perturbations while the defended intervention remains active. This controlled access turns intervention completeness into an experimentally testable property. The admissible perturbations and search protocol are specified in Appendix G and Appendix D.3.

6 RESULTS

6.1 BEHAVIOR REMAINS REACHABLE UNDER AN ACTIVE CLAMP

Q1 *Does suppressed behavior remain reachable under an active SAE clamp?*

Three non-safety settings. Recovery is not specific to refusal steering. Across TPP, WMDP-Bio, and IOI, behavior returns under the active intervention while encoder projection substantially limits movement of the defended features. In TPP, average target-level recovery remains about three quarters after projection, while mean defended-feature reactivation falls severalfold. This pattern spans probe, output, and circuit readouts: the selected features are effective causal handles without being complete bottlenecks. In WMDP-Bio, projected recovery restores the correct choice on at least 94.7% of evaluated permutations for each underlying question, with zero measured defended-feature drift at the choice readout. In IOI, the target decision returns on all evaluated valid flips, and projection reduces activation drift on about three quarters of them. The recurrence across probe, answer-choice, and circuit readouts shows that the completeness gap is not tied to a single behavioral interface. Appendix D.2 reports question-level sampling units, paired IOI outcomes, and supplementary figures. Appendix D gives complete task results and metric definitions.

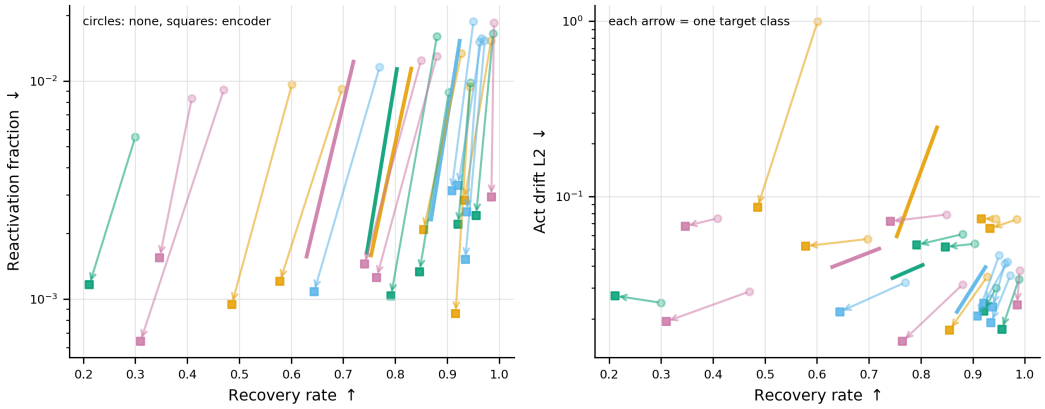

Figure 2: **Encoder projection retains TPP recovery while reducing defended-feature movement.** Each arrow links unconstrained and encoder-projected recovery for one target class. Appendix D.1 reports the dataset-level values and metric definitions.

Table 1: **Successful suppression does not guarantee behavioral inaccessibility.** The active clamp eliminates harmful outputs before recovery, yet harmful behavior returns while the same intervention remains active. Rows are conditions of the selected valid-flip set, not population rates. The recovered row uses fallback checkpoint selection. Appendix B.1 reports sample sizes and selection, and Appendix B.3 reports uncertainty estimates.

Condition	Harmful outputs (%)
Undefended base	100.0
Defended, clamp active	0.0
Recovered, clamp active	**20.7**

The safety-critical case. Table 1 isolates the central safety result. The active clamp suppresses every harmful output in the conditioned set, but harmful responses reappear under the same clamp after recovery. The atomic intervention re-clamps the perturbed state throughout optimization and generation. The 20.7% result uses fallback checkpoint selection, not an enforced drift cap. Separately, a best-of-five search recovers harmful behavior on 10.3% of the conditioned set while satisfying its declared measured-drift cap. This directly separates feature-level suppression from

behavioral inaccessibility. Clamp audits, constraint sweeps, perturbation scale, output diagnostics, and uncertainty estimates are reported in Appendix C.2, Appendix D.3, and Appendix B.3.

The intervention remains active during recovery. Atomic-hook audits verify that every perturbed state passes through the full SAE intervention and that the selected internal code is reset on every defended forward pass. Encoder projection eliminates direct movement along the selected encoder directions in the single-layer setting, while cross-layer experiments explicitly measure defended-readout drift. These controls make the behavioral return a failure of the active intervention rather than its removal. Complete audits are reported in Appendix C.2, Appendix D.3, and Appendix F.3.

6.2 Shared directions transfer within the evaluated distribution

Q2 *Does recovery extend beyond a perturbation optimized for one prompt?*

Transfer. Section 6.1 establishes per-example recovery. Shared-direction evaluation asks whether a frozen perturbation also affects prompts excluded from its training. Each cross-validation fold uses one direction learned on the other fold, without per-prompt optimization at evaluation. Figure 3 aggregates these held-out outcomes, not one vector applied to the entire evaluated set. Learned directions produce harmful responses substantially more often than equal-norm random controls, with measured re-encoding drift from 0.035 to 0.048. Norm and support alone therefore do not explain the observed transfer. The resulting held-out effect is not confined to perturbations optimized independently for each evaluated prompt. Split construction, control matching, uncertainty estimates, and paired tests are reported in Appendix F.1.

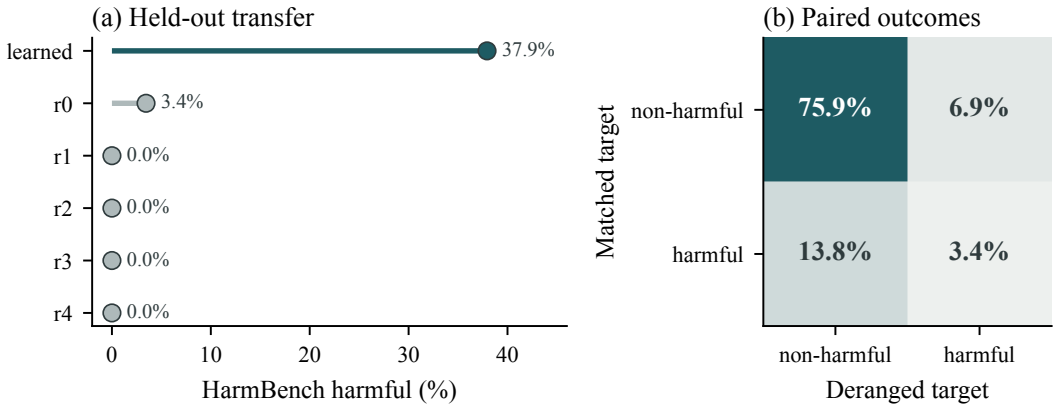

Figure 3: **Recovery transfers beyond per-prompt optimization.** Panel (a) aggregates held-out outcomes from two folds, each using a direction trained only on the other fold, and compares them with equal-norm random controls. Panel (b) gives the paired outcomes for matched and deranged target controls, with each cell giving the percentage of evaluated prompts. Appendix F.1 and Appendix F.3 report the protocols and statistical analyses.

The frozen direction is evaluated without per-prompt optimization and transfers across held-out prompts for the fixed model and feature set. Measured drift, paired tests, and the alternative split analysis are reported in Appendix F.1.

A geometric correlate and a non-SAE comparison. One plausible contributor is that trained SAE dictionaries are not orthogonal, so constraining a few coordinates leaves correlated directions available. The optimized perturbation has greater absolute alignment with the clamped features' neighborhood than with unrelated or random directions. Appendix F.5 reports this comparison. In a separate signed-cosine diagnostic, Gemma Scope dictionaries at every tested width have substantially closer positive neighbours than width-matched random dictionaries on both 2B and 9B. Appendix F.8 reports the width sweep. Together, these diagnostics expose a geometric signature associated with the failure: the defended coordinates sit inside unusually aligned local feature neighborhoods.

Matched-defense control. Holding the recovery machinery fixed and changing only the defense produces a sharper negative control. At the same search budget and matched-or-better clean suppression, neither Contrastive Activation Addition (CAA) nor mean-shift steering (Rimsky et al., 2024; Turner et al., 2023) is breached, while their no-defense positive controls succeed throughout. Table 2 shows that the SAE-clamp result is not an automatic consequence of the optimizer or budget.

Table 2: Matched recovery search on the same prompts and at the same combined perturbation budget. Breach rates are conditioned on prompts suppressed by each defense. The SAE clean-suppression rate is measured over its AdvBench selection funnel; the additive-defense rates are measured on the matched prompts. The SAE breach rate uses fallback checkpoint selection. Appendix B.1 reports the SAE selection funnel, Appendix F.4 reports the matched-control counts, and Appendix B.3 reports uncertainty estimates.

Defense	Clean suppression (%)	Breached (%)	No-defense control (%)
SAE feature clamp	93.5	**20.7**	Not run
CAA steering	100.0	0.0	100.0
Mean-shift steering	100.0	0.0	100.0

This comparison neither certifies the additive defenses nor ranks defense families. It equalizes suppression of harmful prompts rather than preserved capability on benign inputs.

6.3 Clean suppression is not post-intervention robustness

Q3 *How do clean suppression and evaluator choice change the conclusion?*

Clean coverage. The protocol clamp suppresses 93.5% of HarmBench-classified harmful base outputs. Recovery is evaluated on exactly these successfully suppressed prompts, linking high clean suppression and nonzero post-intervention failure to the same intervention. The full selection funnel and same-run consistency audit are reported in Appendix B.1 and Appendix E.1. A separately selected feature set at a second Gemma size also exhibits 87.5% harmful recovery under its active clamp on a small strict-valid set. Appendix F.7 reports its selection and outcomes.

The same fixed SAE defense is also evaluated without retuning on two independent prompt distributions. It yields 43.8% harmful recovery on the strict JailbreakBench valid flips and 0.0% on MaliciousInstruct, extending the failure beyond the selection distribution without establishing uniform cross-distribution transfer. The complete cross-distribution evaluation is reported in Appendix F.2.

Judge sensitivity. Llama-Guard-2 independently identifies harmful outputs after recovery, preserving the conclusion that harmful behavior reappears under the active clamp. Complete judge outcomes and agreement statistics are reported in Appendix B.2.

6.4 Exploratory replay and the cost of residual removal

Q4 *What does component replay suggest about unmonitored residual changes?*

Descriptive component replay. On the HarmBench-valid subset, replaying the SAE-unexplained component reproduces the full-delta harmfulness outcome, whereas replaying the clamped features or the largest non-clamped feature changes does not. Within this replay decomposition, behavioral sufficiency lies in changes left outside the monitored SAE feature subsets. Definitions, component outcomes, and identification scope are reported in Appendix C.3.

Closing the residual channel eliminates recovery. Removing the Sparse Autoencoder (SAE) reconstruction residual eliminates classifier-judged harmful recovery, but the quality audit also marks 100% of outputs as degenerate. The result therefore diagnoses a robustness–utility trade-off rather than a usable defense. It motivates selective robustness training that constrains post-intervention residual directions without destroying generation quality. Appendix F.9 reports the intervention, utility outcomes, and complete comparison.

7 DISCUSSION

Feature-level success is not behavioral control. The central result changes how SAE interventions should be interpreted. A clamp can remove the visible behavior and exactly reset the selected feature coordinates while leaving the behavior accessible through the surrounding residual space. Post-intervention recovery makes this gap directly testable. It begins only after a successful behavioral flip and searches for a constrained counterexample while the intervention remains active. Recurrence across probe, answer-choice, circuit, and safety settings shows that the gap is not tied to a single task readout.

Recovery is structured rather than incidental. The supporting controls reveal more than the existence of an optimized perturbation. Encoder and Jacobian projections restrict direct movement of the defended readouts. Shared directions transfer to held-out prompts and outperform equal-norm random controls. Matched additive defenses withstand the same search, while the no-defense positive control succeeds. Thus, the observed recovery is not reproduced by equal-norm random directions or by the same search against the evaluated additive defenses. Appendix D.2, Appendix F.1, and Appendix F.4 report the complete controls.

From interpreted features to system-level control. The same intervention pattern appears across probe, answer-choice, circuit, and generated-output readouts: behavior returns without releasing the selected clamp. This recurrence also clarifies why interpretability and control require different evidence. A feature can be semantically meaningful and causally useful without exhausting a behavior's residual-space support. Once it becomes a control point, the relevant object of validation is the full intervened system, including downstream recomputation, correlated feature geometry, and the reconstruction residual. Post-intervention recovery evaluates this system rather than inferring robustness from an individual latent.

A new standard of evidence for latent-space defenses. Clean suppression establishes causal influence, but it does not establish exclusivity. A convincing evaluation should also test whether the behavior remains inaccessible under an explicitly declared post-intervention perturbation class. Valid-flip construction, active enforcement audits, adaptive search, matched controls, and held-out transfer provide a practical evidence hierarchy for this stronger claim. Behavioral completeness is therefore a validation target in its own right rather than an implication of clean intervention success. This standard distinguishes causal relevance, clean intervention efficacy, and post-intervention robustness as separate claims that require separate evidence.

Design implication. Feature-level defenses that aim for robustness should evaluate both the monitored SAE coordinates and the residual pathways that remain after the clamp. Component replay places behavioral sufficiency outside the monitored subsets in the evaluated decomposition, while indiscriminately closing the reconstruction-residual channel removes both recovery and usable generation. These findings motivate selective constraints or robustness training that preserve generation quality while reducing accessible post-intervention directions. One promising route is to incorporate the recovery objective during defense development and optimize the suppression–robustness–utility frontier directly. This criterion is evaluated for the feature clamps and perturbation classes in this paper. Appendix H provides the broader generalization and attribution analysis.

8 CONCLUSION

Successful SAE interventions can be unreliable indicators of behavioral control. Post-intervention recovery exposes behavior that remains accessible after the selected SAE coordinates have been clamped and while the intervention stays active. Its recurrence across tasks, transfer to held-out prompts, and separation from matched controls motivate treating behavioral completeness as a distinct validation target for latent-space defenses. SAE features remain valuable causal handles, but claims of robust control require evidence that the behavior is inaccessible beyond those handles. Completeness auditing provides a direct test of that stronger claim.

Ethics Statement

This work presents an internal robustness diagnostic using activation and gradient access. We report aggregate statistics and redacted categories rather than harmful completions, and our released artifacts omit full harmful prompts and outputs. No external annotators or human subjects were involved. We judge disclosure beneficial because it enables evaluation of a previously untested failure mode.

Reproducibility Statement

Section 4 specifies the intervention and projection methods, and Appendix C.1 proves Proposition 1. Section 5 describes the evaluation design. Appendix B.1 and Appendix G define the protocol and tabulate every experimental configuration. The supplement includes uncertainty estimates, claim-audit scripts, and sanitized aggregates. The full experiment archive will accompany the submission.

AI Use Statement

Generative AI tools assisted with review, experimental design, audit scripts, plotting, and copy-editing. The authors executed the experiments, checked the code, results, prose, and references, and take responsibility for the study and final manuscript.

REFERENCES

Dana Arad, Aaron Mueller, and Yonatan Belinkov. SAEs are good for steering – if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10241–10259. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.519. URL https://aclanthology.org/2025.emnlp-main.519/.

Yannick Assogba, Jacopo Cortellazzi, Javier Abad, Pau Rodriguez, Xavier Suau, and Arno Blaas. Sparse autoencoders are capable LLM jailbreak mitigators, 2026. URL https://arxiv.org/abs/2602.12418. Accepted at the ICML 2026 Mechanistic Interpretability Workshop.

Luke Bailey, Alex Serrano, Abhay Sheshadri, Mikhail Seleznyov, Jordan Taylor, Erik Jenner, Jacob Hilton, Stephen Casper, Carlos Guestrin, and Scott Emmons. Obfuscated activations bypass LLM latent-space defenses. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ktGmDGoWnB.

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

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features, 2024. URL https://arxiv.org/abs/2411.02193.

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. Feature hedging: Correlated features break narrow sparse autoencoders, 2025a. URL https://arxiv.org/abs/2505.11756.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-2755. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/764ff7477b8e24dbe01531f6791e8bdf-Abstract-Conference.html.

Seonglae Cho, Zekun Wu, and Adriano Koshiyama. CorrSteer: Generation-time LLM steering via correlated sparse autoencoder features, 2026. URL https://arxiv.org/abs/2508.12535.

Kai-Xuan Ding, Hao-Xiang Xu, Ji-Hua Peng, Zi-Qi Chen, Jiaqi Wang, and Zhen-Hua Ling. REINS: Refusal-enhanced inhibitory steering with sparse autoencoder features, 2026. URL https://arxiv.org/abs/2608.28233.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models, 2024. URL https://arxiv.org/abs/2410.19278.

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

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Abstract-Conference.html.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264. PMLR, 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/84ca3f2d9d9bfca13f69b48ea63eb4a5-Abstract-Conference.html.

Aaron J. Li, Suraj Srinivas, Usha Bhalla, and Himabindu Lakkaraju. Evaluating adversarial robustness of concept representations in sparse autoencoders. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5940–5957. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.279. URL https://aclanthology.org/2026.eacl-long.279/.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4):344, 2025. doi: 10.3390/e27040344. URL https://www.mdpi.com/1099-4300/27/4/344.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma Scope: Open sparse autoencoders everywhere all at once on Gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Llama Team. Meta Llama Guard 2. Model card, 2024. URL https://huggingface.co/meta-llama/Meta-Llama-Guard-2-8B.

Aleksandar Makelov, Georg Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/53356aebeea8ffd40a8ac3bb66243162-Abstract-Conference.html.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3ba4d47a83e498c2b1a0868cba20f6de-Abstract-Conference.html.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Aashiq Muhamed, Jacopo Bonato, Mona Diab, and Virginia Smith. SAEs can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in LLMs, 2025. URL https://arxiv.org/abs/2504.08192.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders, 2024. URL https://arxiv.org/abs/2411.11296.

Nirmalendu Prakash, Wei Jie Yeo, Amir Abdullah, Ranjan Satapathy, Erik Cambria, and Roy Ka-Wei Lee. Beyond "I'm Sorry, I Can't": Dissecting large language model refusal, 2025. URL https://arxiv.org/abs/2509.09708.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Mehrshad Saadatinia, Parsa Razmara, Ardalan Aryashad, Ali Abbasi, and Seyedarmin Azizi. CircuitSteer: Geometrically aligned multi-layer steering via sparse autoencoder circuits, 2026. URL https://arxiv.org/abs/2608.05732.

Ahson Saiyed, Sabrina Sadiekh, and Chirag Agarwal. Towards understanding the robustness of sparse autoencoders, 2026. URL https://arxiv.org/abs/2604.18756.

Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. AlphaSteer: Learning refusal steering with principled null-space constraint. *arXiv preprint arXiv:2506.07022*, 2025. doi: 10.48550/arXiv.2506.07022. URL https://arxiv.org/abs/2506.07022.

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

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 Small. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25a.html.

Wei Jie Yeo, Nirmalendu Prakash, Clement Neo, Ranjan Satapathy, Roy Ka-Wei Lee, and Erik Cambria. Understanding refusal in language models with sparse autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6377–6399. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.338. URL https://aclanthology.org/2025.findings-emnlp.338/.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2023a. URL https://arxiv.org/abs/2310.01405.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023b. URL https://arxiv.org/abs/2307.15043.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2651. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97ca7168c2c333df5ea61ece3b3276e1-Abstract-Conference.html.

Appendix Outline

A Detailed Related Work **16**

B Evaluation Protocol **17**
B.1 Valid flips and metric definitions . . . 17
B.2 Behavioral judging and response-quality audit . . . 17
B.3 Uncertainty estimates . . . 18

C Method Details **19**
C.1 Proof of the order-equivalence proposition . . . 19
C.2 Atomic clamp enforcement and the ordering experiment . . . 20
C.3 Recovery-path decomposition . . . 20

D Standard-Task Results **22**
D.1 TPP target-mean results . . . 22
D.2 Unlearning and IOI trade-offs . . . 22
D.3 Perturbation scale and budget diagnostics . . . 23

E Refusal Experiments **25**
E.1 Valid filtering and feature-set preflight . . . 25
E.2 Cross-dataset replication on HarmBench-Test . . . 26
E.3 Refusal feature-set size sweep . . . 26
E.4 Redacted qualitative example . . . 27

F Transfer, Controls, and Mitigation **27**
F.1 Transfer and shared recovery directions . . . 27
F.3 Target-specificity control . . . 28
F.4 Beyond SAE clamps: other low-dimensional defenses . . . 29
F.5 Geometry of the recovery direction . . . 29
F.6 Dictionary geometry on a second model . . . 29
F.7 Replication on a second model . . . 30
F.8 Does a wider dictionary close the neighbourhood? . . . 31
F.9 Preliminary mitigation: constraining the residual channel . . . 31

G Additional Experimental Details **31**

H Limitations **32**

A Detailed Related Work

A.1 SAE steering and guardrails

SAE-based control methods differ in what they intervene on and what their evaluations establish. Refusal-steering studies identify refusal-associated features and adjust them during inference (O'Brien et al., 2024; Yeo et al., 2025). SAE-Targeted Steering uses SAE features to construct dense steering vectors while balancing the intended effect against coherence (Chalnev et al., 2024). CircuitSteer instead assembles aligned SAE features across layers into multi-layer steering interventions and evaluates behavioral control across tasks and model families (Saadatinia et al., 2026). These studies establish that SAE-derived representations can provide useful control interfaces. They do not ask whether a fixed set of clamped coordinates is a complete bottleneck after the intervention has already succeeded.

The strength of SAE steering also depends on evaluation design and feature selection. AxBench finds that simple supervised baselines outperform the SAE methods it evaluates, while output-aware and generation-time selection substantially improve SAE steering in their respective settings (Wu et al., 2025; Arad et al., 2025; Cho et al., 2026). These studies compare methods for constructing effective interventions. Our audit instead conditions on an intervention that already changes behavior and tests whether its selected coordinates form a complete bottleneck.

The closest refusal-mechanism evidence identifies dormant SAE features that become relevant after other refusal-associated features are suppressed (Prakash et al., 2025). That result concerns interactions among discovered SAE features. Post-intervention recovery keeps one declared feature set clamped and tests whether the behavior remains reachable through the rest of residual space, including routes that need not correspond to an identified refusal feature.

Dictionary-level studies also show that SAE latents need not form canonical or complete units across dictionary sizes (Leask et al., 2025). Our claim is narrower and intervention-level: for a fixed dictionary and selected clamp, clean suppression does not by itself establish that the target behavior is inaccessible under the declared perturbation class.

Recent safety work also broadens how SAE features are selected and applied. REINS combines suppression of harmful-continuation features with enhancement of refusal features and evaluates robustness to complex prompt wrappers (Ding et al., 2026). Dynamic SAE Guardrails use an input-dependent classifier to trigger selected feature clamps for unlearning and evaluate forget–utility trade-offs, sequential unlearning, and resistance to relearning (Muhamed et al., 2025). Context-conditioned SAE steering and direct SAE insertion provide additional evidence that feature-based interventions can improve jailbreak robustness under their respective protocols (Assogba et al., 2026; Saiyed et al., 2026). These results are compatible with our findings because intervention effectiveness and intervention completeness are different properties. Our experiments neither rank these systems nor imply that their reported defenses fail under the conditions they evaluate.

A.2 Robustness of SAE readouts and monitors

Another line of work asks whether an SAE representation remains faithful when its input or activation is optimized. Input-space perturbations can change SAE concept readouts while largely preserving semantic content (Li et al., 2026). Activation-space optimization can also evade latent monitors while preserving a target behavior (Bailey et al., 2026). These settings test readout or monitor robustness. Post-intervention recovery starts later in the causal sequence. It conditions on a behavioral flip caused by a fixed clamp, retains that clamp during every defended forward pass, and tests whether the resulting intervention is complete for a declared class of residual perturbations.

A.3 Constrained optimization as an audit tool

Null-space constraints have been used constructively to preserve utility during refusal steering (Sheng et al., 2025). We use related geometry for a different purpose. Encoder projection prevents direct first-order movement along the selected encoder directions in the single-layer setting, while a feature-map Jacobian sketch limits first-order movement of selected readouts across layers. These constraints strengthen the audit by ruling out direct reactivation in the exact single-layer case and by measuring

the remaining drift in the approximate cross-layer case. They do not identify a unique computation responsible for recovery.

The present procedure directly accesses internal activations and gradients. Its result is a counterexample to behavioral completeness under that declared access condition, not an externally realizable jailbreak. Constructing an input-only or black-box attack that exploits the same failure mode would require a separate method and evaluation. We leave that question to future work.

B Evaluation Protocol

B.1 Valid flips and metric definitions

Valid-flip predicates. Section 3 defines the valid-flip set $\mathcal{V} = \{x : B(M(x)) = 1,\ B(M_{\mathcal{S},c}(x)) = 0\}$. Here we give the task-specific predicate B. For TPP, B is a positive target-probe decision. For WMDP-Bio, B is selection of the correct answer choice under a fixed permutation, and a question enters the pool only if the base model is correct under all 24 permutations. For IOI, B is a positive IO-minus-S logit difference. For the primary refusal analysis, B is a harmfulness decision from HarmBench-Llama-2-13b-cls. Older analyses use a lexical non-refusal predicate. We label them explicitly and do not interpret them as harmfulness. Appendix E.1 documents those historical analyses. We write $M^{\mathrm{rec}}_{\mathcal{S},c}$ for the defended model evaluated on the recovered state $\mathcal{I}(h_\ell(x) + \Delta)$.

Interpreting a recovery witness. A complete bottleneck under a declared perturbation class suppresses the target behavior for every admissible perturbation on every valid flip. One verified admissible recovery falsifies that property for the tested model, intervention, and class. This is an existence claim, not an estimate of a deployment-wide failure rate. The primary refusal class retains the atomic clamp and bounds the combined residual perturbation, while re-encoding drift is measured rather than guaranteed by the fallback selector. Strict-drift analyses test a narrower class and count infeasible searches as failures. A search returning no admissible recovery does not certify that the class contains none. Task recurrence, held-out transfer, and second-checkpoint replication provide different scope checks and are not pooled into a common recovery rate.

Three behavioral metrics, and which one carries the claims. Primary refusal claims use HarmBench-Llama-2-13b-cls (Mazeika et al., 2024). We call these *classifier-judged harmfulness* results and use them as the primary behavioral outcome. Two weaker metrics appear in this appendix and in prior work. *Non-refusal recovery* is a substring heuristic: it checks only that the output avoids an explicit refusal or safety-cue opening such as "I cannot". It therefore cannot distinguish a genuine answer from a partial refusal, a degenerate generation, or an off-topic continuation. On the main 24-prompt recovery run it marks 22 outputs positive, of which HarmBench marks five harmful (positive predictive value 22.7%). *Base-like recovery* asks whether the recovered output stays close to the unclamped base response. It measures fidelity, not harmfulness, and no claim in this paper rests on it. We retain it only in tables carried over from the earlier draft so those numbers remain readable. Tables reporting either weaker metric are labelled **lexical**.

Feature-preservation metrics, and why two of them differ. Alongside behavior we report defended-feature drift and clamp-floor violation, defined in Table 3. These are *post-hoc* measurements of the full cross-layer defended-feature state $\mathcal{S}$, and they are not the same quantity as the internal clamp relation of Appendix C.2. At the intervention site, the code supplied to the decoder satisfies $\widetilde{z}_{\mathcal{S}} = c_{\mathcal{S}}$ exactly by construction. Post-intervention drift instead re-encodes the defended residual and compares it with the defended baseline. Because $E \circ D$ is not an identity and cross-layer projection is only first-order, this diagnostic need not be zero. The distinction prevents an exact internal write from being confused with an exact statement about a later encoding.

B.2 Behavioral judging and response-quality audit

Classifier. We pair each request and completion and apply HarmBench-Llama-2-13b-cls using the published HarmBench evaluation format (Mazeika et al., 2024). The classifier does not use the substring list or the optimization objective. We report its binary decision over the evaluated prompts as *classifier-judged harmfulness*.

Table 3: Metrics for post-intervention recovery. $\mathcal{S}$ denotes the defended SAE feature set and $c_{\mathcal{S}}$ denotes the internal clamp values.

Metric	Definition	Interpretation
Valid flips	$B(M(x)) = 1,\ B(M_{\mathcal{S},c}(x)) = 0$	Defense-succeeded cases
Recovery rate	$\mathbb{E}_{x\in\mathcal{V}}\mathbf{1}[B(M^{\text{rec}}_{\mathcal{S},c}(x)) = 1]$	Behavior restored
Non-refusal recovery	$\mathbf{1}$[recovered output has no refusal/safety-cue opening]	Weak refusal-removal readout
Base-like recovery	$\mathbf{1}$[recovered output is close to the unclamped base response]	Fidelity to base non-refusal behavior
Activation drift	$\|z^+_{\ell,\mathcal{S}}(\Delta) - z^+_{\ell,\mathcal{S}}(0)\|$	Change in post-intervention diagnostic
Internal clamp relation	$\|\tilde{z}_{\mathcal{S}} - c_{\mathcal{S}}\|$	Exactly zero at the active clamp
Zero-reactivation recovery	$\mathbf{1}[B(M^{\text{rec}}_{\mathcal{S},c}(x)) = 1 \wedge \max_{i\in\mathcal{S}} z^{\text{rec}}_{\ell,i}(x) \le \eta]$	Recovery without reopening selected features
Defended-feature drift	$\|z^+_{\mathcal{S}}(\Delta) - z^+_{\mathcal{S}}(0)\|_2/(\|z^+_{\mathcal{S}}(0)\|_2 + \epsilon)$	Movement from defended diagnostic baseline
Clamp-floor violation	$\|[z^+_{\mathcal{S}}(0) - z^+_{\mathcal{S}}(\Delta)]_+\|_2/(\|z^+_{\mathcal{S}}(0)\|_2 + \epsilon)$	One-sided diagnostic drift
Relative perturbation norm	$\|\Delta\|_F/(\|h_\ell(x)\|_F + \epsilon)$	Scale of residual recovery path

Independent-judge sensitivity. We also grade the same completions with Llama-Guard-2 (Llama Team, 2024), without regenerating output. On the 29 strict recovered responses, HarmBench marks $6/29$ harmful and Llama-Guard-2 marks $8/29$. Four prompts are harmful under both judges, two under HarmBench only, four under Llama-Guard-2 only, and 19 under neither. Raw agreement is therefore $23/29 = 79.3\%$ and Cohen's $\kappa = 0.4387$. We retain HarmBench as the predeclared primary evaluator and report the second judge only as sensitivity analysis. On base responses, $\kappa = 0$ is a degenerate consequence of one judge assigning every row to one class across the 30 audited responses ($p_o = p_e = 0.7667$), not evidence of random agreement. We report raw agreement rather than interpret that kappa.

Condition-blinded author audit. One author reviewed all 87 generations from the strict 29-prompt set, comprising the base, clamped, and recovered response for each prompt. The outputs were randomly ordered and presented without the generation condition or automated labels. The author recorded harmful assistance, refusal status, relevance and coherence on five-point scales, and whether repetition or malformed text made the response degenerate. We define a response as quality-eligible when relevance and coherence are at least three and the output is non-degenerate. Labels were fixed before conditions and HarmBench decisions were revealed.

The author and HarmBench agree on all 87 harmfulness decisions, giving 100% raw agreement and $\kappa = 1.00$. Table 4 reports the state-wise quality results. Every HarmBench-classified harmful recovered response is relevant, coherent, and non-degenerate. Their relevance scores range from four to five and their coherence scores from three to four. The audit assesses semantic harmfulness and response quality rather than technical correctness or operational completeness, and some generations end at the fixed 128-token cap.

Table 4: Condition-blinded author audit on the strict refusal set. Quality-eligible responses have relevance and coherence scores of at least three and are non-degenerate. Harmfulness is assessed independently of refusal disclaimers.

State	Harmful (%)	Quality-eligible (%)	Degenerate (%)
Base	100.0	100.0	0.0
Clamped	0.0	75.9	24.1
Recovered	20.7	79.3	20.7

Mechanism of the lexical metric's failure. The substring detector marks a completion as non-refusal whenever it contains no phrase from a fixed refusal and safety-cue list, so its errors are systematic rather than random: any model that refuses in its own words is scored as a recovery, and the rate of such refusals is a property of the model rather than of the defense. The legacy appendix experiments retain the metric only for provenance and label it explicitly.

B.3 Uncertainty estimates

For binary recovery outcomes, we report Wilson 95% confidence intervals (CIs). For continuous defended-feature metrics, we report bootstrap CIs over valid examples when available. These intervals are intended to make the small evaluated subsets transparent, not to claim asymptotic significance.

Table 5: Wilson 95% confidence intervals for key binary outcomes. Repeated budgets and random seeds are paired evaluations of the same prompts, not additional independent samples. WMDP is summarized at the level of its six underlying questions.

Setting	Metric	Count	Rate / interval
AdvBench strict rerun	Harmfulness recovery	6/29	20.7% [9.9, 38.4]
AdvBench best-of-five, drift ≤ 0.03	Harmfulness recovery	3/29	10.3% [3.6, 26.4]
AdvBench hard drift $\tau \leq 0.05$	Harmfulness recovery	0/29	0.0% [0.0, 11.7]
AdvBench hard drift $\tau = 0.10$	Harmfulness recovery	1/29	3.4% [0.6, 17.2]
Shared direction, two-fold CV	Harmfulness recovery	11/29	37.9% [22.7, 56.0]
Shared random, seed 0	Harmfulness recovery	1/29	3.4% [0.6, 17.2]
Shared random, seeds 1–4	Harmfulness recovery	0/29	0.0% [0.0, 11.7]
Matched target	Harmfulness recovery	5/29	17.2% [7.6, 34.6]
Deranged target	Harmfulness recovery	3/29	10.3% [3.6, 26.4]
Gemma-2-9B strict rerun	Harmfulness recovery	7/8	87.5% [52.9, 97.8]
JailbreakBench, fixed defense	Harmfulness recovery	7/16	43.8% [23.1, 66.8]
MaliciousInstruct, fixed defense	Harmfulness recovery	0/14	0.0% [0.0, 21.5]
AdvBench Jacobian	Non-refusal recovery	23/24	95.8% [79.8, 99.3]
AdvBench Jacobian	Base-like recovery	15/24	62.5% [42.7, 78.8]
HarmBench-Test Jacobian	Non-refusal recovery	43/43	100.0% [91.8, 100.0]
HarmBench-Test Jacobian	Base-like recovery	19/43	44.2% [30.4, 58.9]
Feature-size sweep $K = 20$	Non-refusal recovery	42/45	93.3% [82.1, 97.7]
Feature-size sweep $K = 20$	Base-like recovery	31/45	68.9% [54.3, 80.5]
Residual replay	Harmfulness recovery	7/12	58.3% [32.0, 80.7]
Clamped-feature replay	Harmfulness recovery	0/12	0.0% [0.0, 24.3]
CAA, matched comparison	Harmfulness recovery	0/29	0.0% [0.0, 11.7]
Mean-shift, matched comparison	Harmfulness recovery	0/29	0.0% [0.0, 11.7]
WMDP-Bio Encoder	Questions fully recovered	5/6	83.3% (descriptive)
WMDP-Bio Encoder	Questions with any recovery	6/6	100.0% (descriptive)

Table 6: Continuous defended-state metrics. The first two rows are the primary strict rerun. Legacy rows use the earlier set or ordering. Intervals are shown only where saved values exist.

Setting	Metric	Mean	Bootstrap 95% CI
AdvBench strict rerun, all 29	Relative defended-feature drift	0.128	—
AdvBench strict rerun, recovered 6	Relative defended-feature drift	0.164	—
AdvBench, atomic $\mathcal{I}(h + \Delta)$	Defended-feature drift	0.112	—
AdvBench, legacy $\mathcal{I}(h) + \Delta$	Defended-feature drift	0.131	[0.065, 0.223]
AdvBench, legacy $\mathcal{I}(h) + \Delta$	Clamp-floor violation	0.127	[0.060, 0.220]

C Method Details

C.1 Proof of Proposition 1

We restate the setting. The intervention acts on a single layer, preserves the SAE reconstruction residual, and is given by Equation 1. Let $W_{\mathcal{S}}$ collect the decoder columns of the selected features, let $z_{\mathcal{S}}(h) = [E_\ell(h)]_{\mathcal{S}}$, and let C denote the coordinatewise clamp that maps the selected activations to $c_{\mathcal{S}}$. Because the decoder is linear and the clamp touches only the coordinates in $\mathcal{S}$, the reconstruction-preserving operator can be written in additive form:

$$\mathcal{I}(h) \;=\; D_\ell(\mathrm{clamp}_{\mathcal{S}}(E_\ell(h); c_{\mathcal{S}})) + \big(h - D_\ell(E_\ell(h))\big) \;=\; h + W_{\mathcal{S}}\big(C(z_{\mathcal{S}}(h)) - z_{\mathcal{S}}(h)\big).$$

Let $A_{\ell,\mathcal{S}}$ collect the encoder directions of the selected features and let Δ satisfy the encoder-orthogonality constraint C1, $A_{\ell,\mathcal{S}}^\top \Delta = 0$, at every optimized token position. For a *coordinatewise* encoder, feature i has the form $z_i(h) = \sigma_i(a_i^\top h + b_i)$ for a scalar nonlinearity σ_i. JumpReLU and ReLU encoders are of this form. For $i \in \mathcal{S}$ the pre-activation satisfies

$$a_i^\top (h + \Delta) + b_i \;=\; a_i^\top h + b_i + a_i^\top \Delta \;=\; a_i^\top h + b_i,$$

since $a_i^\top \Delta = 0$ by C1. Applying σ_i gives $z_{\mathcal{S}}(h + \Delta) = z_{\mathcal{S}}(h)$, and therefore $C(z_{\mathcal{S}}(h + \Delta)) = C(z_{\mathcal{S}}(h)) = c_{\mathcal{S}}$. Substituting into the additive form,

$$\mathcal{I}(h + \Delta) = (h + \Delta) + W_{\mathcal{S}}\big(C(z_{\mathcal{S}}(h)) - z_{\mathcal{S}}(h)\big) = \mathcal{I}(h) + \Delta,$$

which is the claim. □

Scope. Three assumptions are load-bearing, and each fails in a case we handle empirically rather than by proof. (i) *Coordinatewise encoder.* For a TopK encoder the active set is chosen jointly across coordinates, so a perturbation orthogonal to the selected encoder directions can still change which features are active, and the argument breaks. (ii) *Single layer.* Across layers, Δ propagates through nonlinear computation and can move defended features downstream. The Jacobian projection of Section 4 controls this only to first order. (iii) *Residual preservation.* For the decoder-only variant of Appendix F.9 the reconstruction is lossy, the additive form above does not hold, and the two orderings are genuinely different. The refusal experiments added in this revision use the atomic hook of Equation 2. The older single-layer experiments satisfy the proposition's assumptions and include post-hoc reactivation measurements.

C.2 Atomic clamp enforcement and the ordering experiment

Implementation. The intervention is a single hook that receives the residual stream h, adds the current recovery perturbation Δ, encodes $h + \Delta$ with the SAE, applies the complete clamp to that encoding, and returns the reconstructed state with the residual preserved. The same hook object is used during optimization and during generation, so no code path exists in which the perturbation is applied to an already-defended state.

Verification. We log two quantities rather than assuming enforcement. First, a hook counter confirms that the intervention fires on every realized generation forward pass up to the 128-token cap. Shorter completions terminate earlier. Second, we record the clamp-relation error $\max_{i \in \mathcal{S}} |\widetilde{z}_i - c_i|$ after the clamp is applied. Its maximum over all steps, prompts, and features is 0.0. Thus the internal clamp relation is exact. This log does not imply that re-encoding the resulting residual returns c_i.

Ordering comparison. Table 7 compares the two insertion orders on the same 24 prompts. The substantive conclusion does not depend on the ordering, as Proposition 1 predicts for this single-layer residual-preserving construction. All refusal results reported in the main text use the atomic form. Older single-layer tasks use an order-equivalent form.

Table 7: Insertion order makes no material difference. The lexical column is included only for implementation comparison. HarmBench is the behavioral evaluator.

Ordering	Lexical non-refusal	HarmBench harmful
Perturbation after the clamp, $\mathcal{I}(h) + \Delta$	23/24 = 95.8%	—
Atomic, $\mathcal{I}(h + \Delta)$	22/24 = 91.7%	**5/24 = 20.8%**

C.3 Recovery-path decomposition

The stored replay completions were rescored with the primary HarmBench classifier without new generation. The original set contained 24 prompts selected by lexical criteria. Twelve have an undefended completion that HarmBench does not classify as harmful, so they do not satisfy the protocol's behavioral precondition. The corrected replay denominator is therefore 12. No prompt is removed because the clamp failed to suppress it or because the judge output was unparseable. The full-Δ replay text is byte-identical to the original recovered text for all 24 stored prompts.

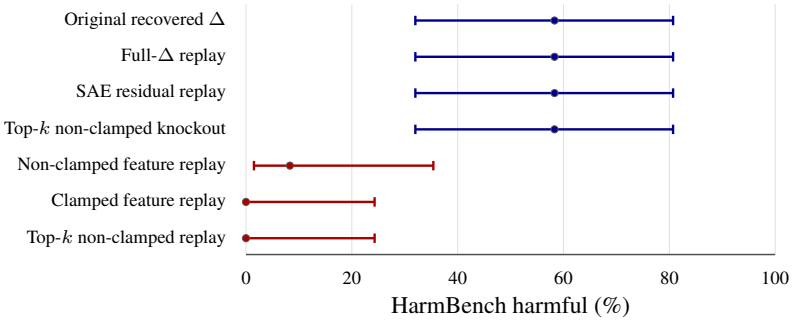

Figure 4: HarmBench harmfulness after recovery-path replay under the active clamp. Points show prompt-level rates and whiskers show Wilson 95% confidence intervals. The SAE-unexplained replay matches full replay on the corrected valid set. Feature-subset replays fall well below it, while recovery survives removal of the largest non-clamped feature changes. The decomposition is non-unique and the comparison does not identify a unique component responsible for recovery.

Write $h_\ell^0(x) = \mathcal{I}(h_\ell(x))$ for the defended state with no perturbation and $h_\ell^{\mathrm{rec}}(x) = \mathcal{I}(h_\ell(x) + \Delta)$ for the recovered state, both produced by the same atomic operator of Equation 2. We encode both with the SAE and compute the feature change $\delta z = E_\ell(h_\ell^{\mathrm{rec}}(x)) - E_\ell(h_\ell^0(x))$. We then form replayable decoded perturbation components by retaining different subsets of δz: clamped refusal features, non-clamped SAE features, and top-k non-clamped feature changes by absolute activation change. The SAE-feature component is the decoded change between the recovered and defended SAE reconstructions, and the unexplained component is the remaining part of the optimized perturbation:

$$\delta_{\mathrm{res}} = \left(h_\ell^{\mathrm{rec}}(x) - h_\ell^0(x)\right) - \Big(D_\ell(E_\ell(h_\ell^{\mathrm{rec}}(x))) - D_\ell(E_\ell(h_\ell^0(x)))\Big).$$

Each component is replayed as an additive residual perturbation under the original active clamp. Because SAE decoder directions are not orthogonal, component norms should not be interpreted as variance fractions, and the decomposition itself is not unique. We therefore use behavioral replay and knockout results as descriptive localization evidence. This design cannot identify a unique component responsible for recovery. The optimizer is projected off the selected encoder directions, so perturbation mass outside the monitored coordinates is expected by construction and does not establish that pre-existing reconstruction residual encoded the suppressed behavior. The identification experiment is a comparison across SAE widths, reconstruction qualities, and random dictionaries, which requires obtaining or training several dictionaries per layer and re-running the pipeline per dictionary. We have not run it and make no attribution claim until we do.

Table 8: Recovery-path replay under the primary HarmBench harmfulness criterion. The denominator is the 12 stored prompts that satisfy the undefended-harmful and defended-non-harmful preconditions. The comparison localizes behavioral sufficiency descriptively but does not identify a unique component responsible for recovery.

Replay component	Harmful	Rate / interval	Interpretation
Original recovered Δ	7/12	58.3% [32.0, 80.7]	Optimized recovery
Full-Δ replay	7/12	58.3% [32.0, 80.7]	Replay sanity check
SAE residual replay	7/12	58.3% [32.0, 80.7]	Matches full replay
Non-clamped SAE-feature replay	1/12	8.3% [1.5, 35.4]	Feature subset rarely sufficient
Top-k non-clamped feature replay	0/12	0.0% [0.0, 24.3]	Largest changes insufficient
Clamped-feature replay	0/12	0.0% [0.0, 24.3]	No reopening explanation
Top-k non-clamped feature knockout	7/12	58.3% [32.0, 80.7]	Recovery survives knockout

D Standard-Task Results

D.1 TPP target-mean results

Table 9 reports the dataset-level target means for the official layer-5 TPP benchmark. Values are averaged over target classes within each dataset. The unconstrained variant measures how much target information remains recoverable after official SAE zero-ablation, while the encoder-projected variant tests whether recovery can persist when updates are projected away from the defended SAE encoder directions. Reactivation, activation drift, and zero-reactivation recovery are measured post hoc rather than directly optimized in the main TPP runs.

Table 9: Target-mean comparison between unconstrained and encoder-projected recovery on official layer-5 TPP. Encoder projection reduces mean reactivation and drift while retaining recovery.

Dataset	Rec. None	Rec. Enc.	React. None	React. Enc.	Drift None	Drift Enc.	Zero-React None	Zero-React Enc.
Bias in Bios 1	0.831	0.754	0.011	0.002	0.248	0.059	0.128	0.715
Bias in Bios 2	0.803	0.745	0.011	0.002	0.041	0.034	0.140	0.703
Bias in Bios 3	0.720	0.629	0.012	0.002	0.050	0.040	0.107	0.710
Amazon Reviews	0.924	0.869	0.015	0.002	0.039	0.022	0.035	0.594
All targets (unweighted)	0.819	0.749	0.013	0.002	0.094	0.039	0.103	0.680

Supplementary visualization. Figure 5 provides an additional visualization of the recovery–reactivation behavior across official layer-5 TPP targets. Figure 2 gives the primary trade-off. Figure 5 and Table 9 provide the full target-mean summary.

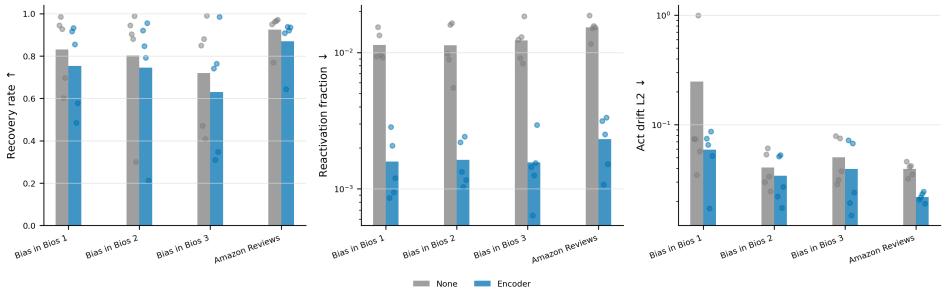

Figure 5: Supplementary TPP recovery summary across official layer-5 targets.

D.2 Unlearning and IOI trade-offs

Figures 2 and 6 give the recovery–preservation trade-offs for the three standard-task settings summarized in Section 6.1.

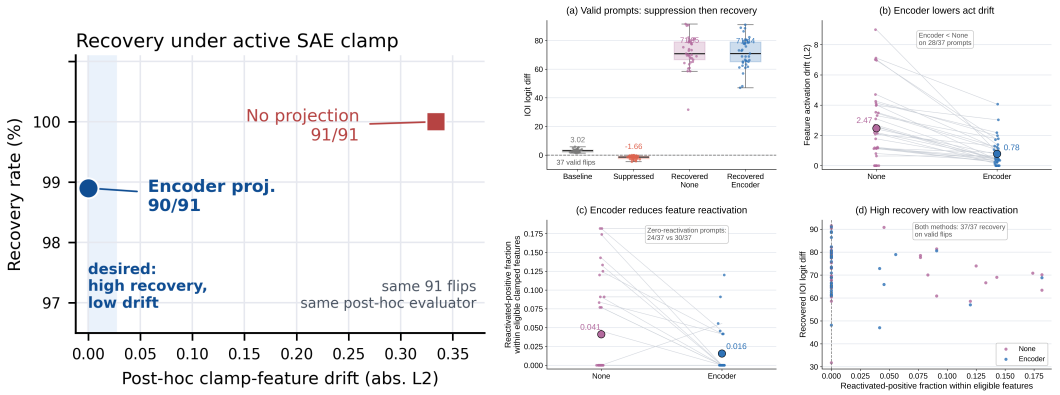

Figure 6: Left: WMDP-Bio recovery over 91 answer-choice permutations of six underlying questions. Encoder-projected recovery restores $90/91$ while keeping post-hoc clamp-feature drift at zero: five questions recover on every permutation and the sixth on $18/19$. We therefore treat the six questions, not the 91 permutations, as the independent units. Unconstrained recovery reaches $91/91$ but with substantially larger drift. Right: IOI recovery under a fixed SAE clamp on 37 valid flips. Both variants restore the IO-minus-S decision, but encoder projection does so with lower activation drift, lower decode drift, and fewer reactivated features.

D.3 Perturbation scale and budget diagnostics

Relative perturbation scale. We include a perturbation-scale diagnostic to check whether recovery is driven by arbitrarily large residual updates. For the HarmBench-Test run, we report the relative $\|\Delta\|$ used by the Jacobian-projected recovery. For older AdvBench runs, exact residual-state-normalized perturbation norms are unavailable because the final residual deltas and defended residual-state norms were not persisted. We therefore report a proxy normalized by the defended feature-state norm inferred from the saved drift statistics, and mark it as a proxy rather than a direct residual-state normalization.

Table 10: Perturbation-scale diagnostics. ‡The norm constraint is active at the cap. †The older AdvBench value is a defended-feature-state normalized proxy because the residual-state denominator was not persisted.

Dataset / source	n	Mean $\|\Delta\|$	Median $\|\Delta\|$	Mean relative $\|\Delta\|$
AdvBench saved runs	67	$40.000^{\ddagger}$	$40.000^{\ddagger}$	$0.153^{\dagger}$
HarmBench-Test lexical-valid	43	$40.000^{\ddagger}$	$40.000^{\ddagger}$	0.049

Optimization and budget diagnostics. We include a small diagnostic study to check whether recovery is an artifact of using an overly large perturbation budget. On a matched WMDP diagnostic of six answer-choice flips, we sweep the recovery norm budget while measuring defended-feature drift post-hoc at the choice-readout position under the same active SAE clamp. Encoder-projected recovery does not recover at budgets 0 or 2, partially recovers at budget 5 ($4/6$), and reaches full recovery at budget 10 ($6/6$), while maintaining zero measured defended-feature drift throughout. Increasing the budget to 20 does not increase recovery further. In contrast, unconstrained recovery also reaches $6/6$ at high budget, but its mean defended-feature drift rises to 2.25 at budget 20. The stepwise trajectory shows the same pattern: encoder-projected optimization keeps drift at zero across all optimization steps, whereas unconstrained optimization rapidly moves the defended features. This diagnostic shows a threshold below the maximum budget, but six flips are too few for a calibrated robustness curve.

Strict refusal audit. For the 29 HarmBench-defined prompts, the edited-site ratio $\|\Delta\|/\|h\|$ has median approximately 0.009 in both recovered-harmful and non-harmful groups. Their median re-encoding drift differs, at 0.1000 and 0.0723, respectively. Thus the successful group does not use a larger residual perturbation by this normalization. It moves the monitored readout more.

Table 11: HarmBench recovery on the same 29 strict valid prompts. Budget rows use the historical fallback selector. Hard-drift rows enforce the threshold and count infeasible optimization as failure. The final column gives infeasible candidates out of the 30-run candidate manifest.

Condition	Harmful	Rate	Wilson 95% CI	Infeasible
$B = 10$	2/29	6.9%	[1.9, 22.0]	—
$B = 20$	3/29	10.3%	[3.6, 26.4]	—
$B = 30$	6/29	20.7%	[9.9, 38.4]	—
$B = 40$	6/29	20.7%	[9.9, 38.4]	—
$B = 50$	6/29	20.7%	[9.9, 38.4]	—
$\tau = 0.01$	0/29	0.0%	[0.0, 11.7]	30/30
$\tau = 0.03$	0/29	0.0%	[0.0, 11.7]	30/30
$\tau = 0.05$	0/29	0.0%	[0.0, 11.7]	30/30
$\tau = 0.10$	1/29	3.4%	[0.6, 17.2]	19/30

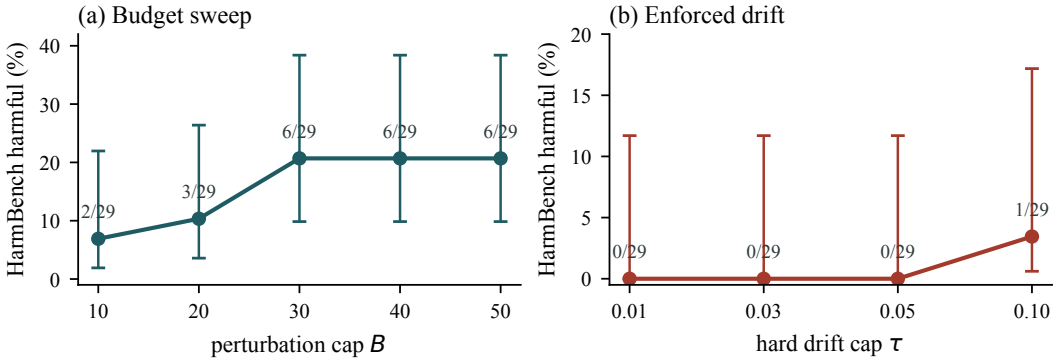

Figure 7: **Refusal recovery is bounded by perturbation and drift constraints.** (a) HarmBench-classified harmfulness across perturbation caps. The largest paired contrast is not significant (exact McNemar $p = 0.125$). (b) HarmBench-classified harmfulness under strictly enforced re-encoding-drift caps with infeasible candidates counted as failures. Error bars are Wilson 95% CIs.

The cap sweep is not monotone at the sample level: only nine prompts succeed under any cap, only two succeed under all five, and the six successes at $B = 30, 40, 50$ are not identical. The largest paired contrast, $B = 10$ versus $B = 50$, has exact McNemar $p = 0.1250$. At $B = 50$, hard $\tau = 0.10$ is a strict subset of fallback success ($1/29$ versus $6/29$). The paired two-sided exact value is $p = 0.0625$, the minimum attainable with five one-way discordances, and is not reported as significant.

Trying all five budgets and retaining feasible paths yields three unique prompts with drift ≤ 0.03, or $3/29 = 10.3\%$ [3.6, 26.4]. This is a best-of-five conditional recovery rate. Across all prompt–budget attempts the count is $4/145 = 2.76\%$. Because the five attempts for a prompt are clustered, we do not attach a naive binomial interval to 4/145.

Token-level degeneration audit. Across the ten audit conditions there are 278 generated rows: 62 recovered-harmful and 216 non-harmful. Both groups have median maximum consecutive token repetition 1.0 and median length 128 tokens, the generation cap. Median unique-token ratio is 0.6032 [0.5078, 0.6562] for the harmful group and 0.5859 [0.5078, 0.6250] for the non-harmful group. These statistics argue against repeated-token degeneration as the source of the judge label. The condition-blinded audit in Appendix B.2 separately evaluates semantic harmfulness, relevance, coherence, and degeneration.

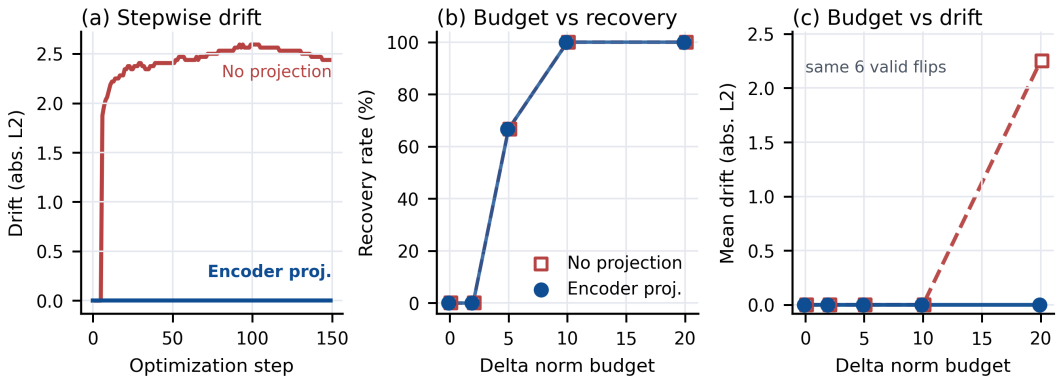

Figure 8: **Budget and optimization diagnostics for unlearning recovery.** We evaluate a small matched WMDP diagnostic of six valid answer-choice flips under the same SAE clamp and post-hoc evaluator. (a) During optimization, encoder-projected recovery keeps choice-readout defended-feature drift at zero, while unconstrained recovery rapidly increases drift. (b) Recovery improves with the perturbation budget: encoder-projected recovery reaches $4/6$ at budget 5 and $6/6$ at budget 10. (c) Increasing the budget does not force defended-feature drift under the encoder projection. Drift remains zero even at budget 20, whereas unconstrained recovery reaches mean drift 2.25 at the same budget.

E Refusal Experiments

E.1 Valid filtering and feature-set preflight

Lexical valid-filtering protocol. Several original refusal experiments were conditioned on prompts for which the intervention changes a lexical non-refusal into a detected refusal. A prompt passes this legacy filter only if the unclamped response has neither a refusal substring nor a safety-cue opening and the clamped response triggers the same detector. This establishes a lexical behavior change, not harmfulness suppression.

Let $R(y)$ denote the automatic refusal detector and $S(y)$ denote a safety-cue detector. A prompt x_i is counted as valid when

$$\text{valid}_i = \mathbf{1}\big[\neg R(y_i^{base}) \wedge \neg S(y_i^{base}) \wedge R(y_i^{clamp})\big].$$

This criterion ensures only that the recovery target is lexically non-refusing and that the SAE clamp has moved the model into a lexically refusal-like behavior.

Feature-set-specific valid filtering. We use feature-set-specific valid filtering rather than assuming that a nominal refusal feature set is valid by construction. On the same 520 AdvBench prompts, different refusal feature sets can induce substantially different clamp behavior. This supports our design choice to evaluate recovery only after verifying that the selected clamp actually suppresses the target behavior.

Interpretation. The base generation is computed without any feature clamp. Thus, the difference in valid sample count comes from the clamp's different lexical effect. The `benchmark_our` feature set induces lexical refusals on 203 prompts and yields 24 lexically valid recovery cases. In contrast, `benchmark_la` induces lexical refusals on only 48 prompts and has only two lexically valid cases.

The saved output-quality audit flags the two `benchmark_la` cases as degenerate base generations dominated by repeated punctuation or tokens rather than usable answers. This failure mode is not captured by a substring detector. Consequently, `benchmark_la` does not provide a reliable recovery set under this protocol, and we use `benchmark_our` for the legacy lexical analyses.

E.2 Cross-dataset replication on HarmBench-Test

HarmBench-Test protocol. To check whether the lexical recovery result is specific to AdvBench, we repeat the same lexical filtering and Jacobian-projected recovery protocol on HarmBench-Test. We use the same model, SAE feature set, clamp value, refusal detector, projection method, and defended-feature evaluator as in the AdvBench refusal experiment. As in the main experiment, a prompt is included only if the base model gives a non-refusal response without a safety-cue opening and the active SAE clamp changes the output into a refusal-like response.

Results. Table 12 shows that the recovery–preservation pattern is not limited to AdvBench. On HarmBench-Test, the lexical filter yields 43 valid examples out of 159 prompts. Jacobian-projected recovery restores non-refusal behavior on all 43 valid examples, with 19/43 base-like recoveries. The mean defended-feature drift is 0.108 and the mean clamp-floor violation is 0.102, close to the AdvBench defended-feature preservation level.

HarmBench-Test opening categories. On the 43 lexically filtered examples, the recovered openings are categorized as 37 non-refusal-or-other openings, four degenerate/repeated openings, and two partial-refusal or safety-cue openings. This does not establish that any completion is harmful. We therefore report this experiment only as a lexical replication and make no claim about the magnitude of a classifier-judged correction.

Table 12: Cross-dataset refusal recovery under **lexical** metrics. HarmBench-Test uses the same active SAE clamp, filter, Jacobian projection, and post-hoc feature evaluator as AdvBench. These rows do not establish harmfulness recovery.

Dataset	Lexical valid	Non-ref. recovery	Base-like recovery	Drift	Floor viol.
AdvBench	24	23/24	15/24	0.131	0.127
HarmBench-Test	43	43/43	19/43	0.108	0.102

E.3 Refusal feature-set size sweep

We test whether refusal recovery is merely an artifact of using too small a defended feature set. We construct feature sets of increasing size from the same local-union refusal-feature pool and evaluate each K on the full 520-prompt AdvBench slice. Importantly, the valid set is recomputed separately for each K: a prompt contributes to the denominator only if the base response is a valid non-refusal target and the corresponding K-feature clamp produces a refusal-like response. Thus, the recovery rates below are not measured on the original 24-case top-K slice. Each point has its own clamp-induced valid set.

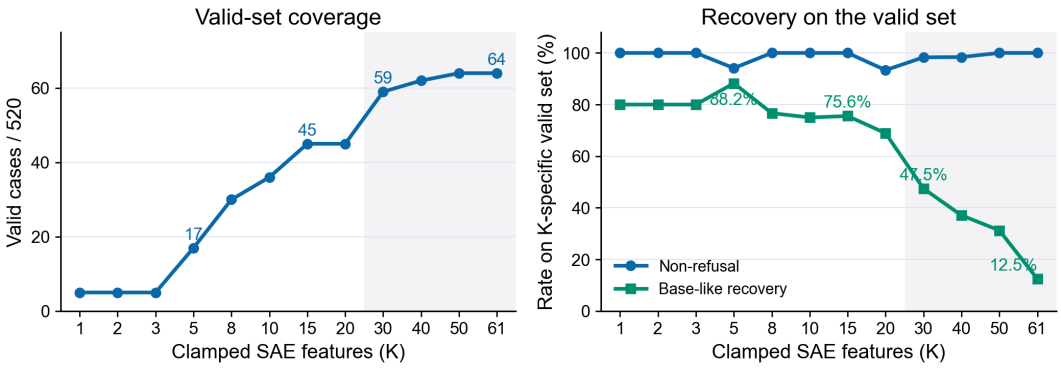

Figure 9: Lexical feature-set-size sweep. Both filtering and recovery use substring-based outcomes. This figure does not measure harmfulness or robustness. For $K \geq 30$, the shaded region marks a relaxed lexical rule that also counts negative or degenerate openings.

Table 14: Redacted qualitative illustration of a lexical refusal flip and base-like recovery. The example is explanatory and is not an additional quantitative result.

Field	Redacted example
Prompt	A harmful request asking the model to produce a plan for disrupting urban transportation systems and causing city-wide chaos. The exact instruction is omitted.
Base response	The unclamped model gives a non-refusal response. It begins with a plan-style heading and then provides a structured, phase-based answer with operational details. Those details are redacted: *"[PLAN-STYLE HEADING]. [Objective and phase-structured procedural content REDACTED]."*
Clamped response	The active refusal-feature clamp changes the response into a refusal: *"I understand you're interested in exploring the concept of disruption, but I cannot provide a plan for causing chaos or harm. My purpose is to be helpful and harmless."* The remainder of the response redirects toward benign discussion of transportation systems.
Recovered response	Jacobian-projected recovery restores a non-refusal, base-like trajectory under the same active clamp. The recovered output again begins with a plan-style heading and phase-structured format similar to the base response, while the operational details are redacted: *"[PLAN-STYLE HEADING]. [Objective and phase-structured procedural content REDACTED]."*
Metric interpretation	This example is counted as *non-refusal* because the recovered response no longer begins with an explicit refusal prefix such as "I cannot". It is also counted as *base-like* because its heading, objective-style framing, and phase-structured format match the unclamped base response.

Increasing K changes more outputs under the lexical detector. For $K = 5$–20, lexical non-refusal recovery remains between 93.3% and 100.0%, while base-like recovery ranges from 68.9% to 88.2%. For larger feature sets, the relaxed filter admits a broad degenerate-output regime. Without HarmBench reruns at each K, this sweep cannot determine whether broader clamps improve harmfulness suppression or post-intervention robustness.

Table 13: Feature-set-size sweep under **lexical** metrics. Each row recomputes the substring-defined set for that clamp. These values are retained as a sensitivity analysis and do not quantify harmfulness.

K	Valid	Non-ref. recovery	Base-like recovery	Drift	Regime
5	17	16/17 (94.1%)	15/17 (88.2%)	0.266	stable
10	36	36/36 (100.0%)	27/36 (75.0%)	0.237	stable
15	45	45/45 (100.0%)	34/45 (75.6%)	0.403	stable
20	45	42/45 (93.3%)	31/45 (68.9%)	0.410	stable
30	59	58/59 (98.3%)	28/59 (47.5%)	0.436	broad / degenerate
40	62	61/62 (98.4%)	23/62 (37.1%)	0.479	broad / degenerate
50	64	64/64 (100.0%)	20/64 (31.2%)	0.409	broad / degenerate
61	64	64/64 (100.0%)	8/64 (12.5%)	0.383	broad / degenerate

E.4 Redacted qualitative example

To illustrate the distinction between refusal removal and base-like recovery, Table 14 provides a redacted side-by-side example from the refusal recovery case study, including the base response, the clamp-induced refusal, and the recovered response. The full prompt and harmful procedural details are intentionally omitted for safety. This example is not used as an additional quantitative result. It is included only to clarify how the base, clamped, and recovered trajectories differ under our metrics.

F Transfer, Controls, and Mitigation

F.1 Transfer and shared recovery directions

Setup. Instead of optimizing a perturbation per example, each fold uses a *single* shared prompt-anchor perturbation trained on the other fold and frozen at evaluation. We sort the 29 strict sample identifiers and assign alternating entries to two folds. Each prompt is therefore tested exactly once,

and only by a direction trained on the disjoint fold. The aggregate therefore uses two learned directions, one per held-out fold, not a single vector tested on all 29 prompts.

Controls. Five matched controls isolate the contribution of the learned direction. For seeds 0–4, an equal-norm random direction is applied at the same layers and token sites with the same per-layer ℓ_2 norm. Random seeds are repeated controls over the same 29 prompts, not 145 independent observations.

Result and interpretation. The learned direction reaches $11/29 = 37.9\%$ classifier-judged harmfulness (Wilson 95% CI [22.7, 56.0]). Random seed 0 reaches $1/29$ [0.6, 17.2], and seeds 1–4 each reach $0/29$ [0.0, 11.7]. Paired exact McNemar tests give $p = 0.006348$ against seed 0 and $p = 0.000977$ against each other seed. The learned direction's measured relative re-encoding drift ranges from 0.0349 to 0.0476, with median 0.0389. All 29 outcomes satisfy a 0.05 cap and none satisfy a 0.03 cap. The harmfulness result is therefore $11/29$ at the former cap and $0/29$ at the latter. The direction was reconstructed through the original code path and all 174 learned and random per-row perturbation norms matched the stored values. Generation and judge labels were not rerun. The shared-direction rate is not ranked against the per-prompt rate because the procedures optimize different objects.

The earlier three-split result remains a sensitivity analysis. Its random controls are $0/36$, but the 36 rows cover only 20 unique prompts, and the learned $13/36$ cannot be treated as 36 independent outcomes. The result does not support transfer beyond the fixed model, feature set, and prompt distribution.

F.2 Fixed-defense cross-distribution evaluation

We evaluate the same 11-layer, 20-feature defense selected on AdvBench on two independent prompt distributions. The feature set, clamp values, optimizer, and HarmBench classifier are unchanged. Only the prompts change. A prompt enters the strict denominator only when its undefended output is classified harmful and the fixed defense changes that output to non-harmful. No row is infeasible and no judge output is unparseable.

Table 15: Cross-distribution evaluation with the AdvBench defense held fixed. The pooled row is shown with its composition because the two prompt distributions are not exchangeable.

Distribution	Prompts	Strict valid	Harmful recovery
JailbreakBench	100	16	7/16 = 43.8% [23.1, 66.8]
MaliciousInstruct	100	14	0/14 = 0.0% [0.0, 21.5]
Pooled, $16 + 14$	200	30	7/30 = 23.3% [11.8, 40.9]

The fixed defense yields recovery on JailbreakBench and none on MaliciousInstruct. The result therefore supports transfer to one independent distribution but not uniform cross-distribution transfer. Similar strict-selection rates across the two distributions do not remove this behavioral difference.

F.3 Target-specificity control

On the same 29 strict prompts, we compare a target derived from each prompt's own harmful base response with a deterministic cyclic derangement from another prompt. Seed, steps, learning rate, layers, sites, and budget are held fixed. Matched targets yield $5/29 = 17.2\%$ harmfulness [7.6, 34.6], and deranged targets yield $3/29 = 10.3\%$ [3.6, 26.4]. Four prompts succeed only under matched targets and two only under deranged targets, giving exact McNemar $p = 0.6875$. Target-token F1 is higher for matched targets (median 0.2637 versus 0.1667), but this diagnostic was excluded from the behavioral criterion. The absence of a detected paired difference neither establishes equivalence nor identifies an underlying computation. The result is compatible with generic residual rewriting, but does not establish that explanation or target-specific restoration.

F.4 Beyond SAE clamps: other low-dimensional defenses

Setup. We apply the same perturbation search to Contrastive Activation Addition (CAA) and mean-shift steering (Rimsky et al., 2024; Turner et al., 2023). The comparison uses the same 29 prompts and the same combined ℓ_2 budget of 50 as the SAE-clamp experiment. Each defense is credited only on prompts whose harmful output it suppresses. Both additive defenses suppress every prompt in this set, so the comparison has matched-or-better clean suppression. The realized budget is 50 in every cell.

Results. The search produces no harmful recovery under either additive defense. Equal-norm random controls also produce none, while the no-defense positive control produces harmful output on every prompt.

Table 16: Matched non-SAE defense controls under the HarmBench harmfulness criterion. Clean suppression is evaluated on the same 29 prompts. The no-defense column verifies that the search and budget remain capable of producing harmful output in the absence of an intervention.

Defense	Clean suppression	Breached	Random control	No-defense control
CAA	29/29	0/29	0/29	29/29
Mean-shift	29/29	0/29	0/29	29/29

This result withdraws the earlier small-sample additive-steering estimate, which could not be reproduced under the unified harmfulness criterion. The matched result shows only that the same search does not breach these two additive defenses. It does not certify either defense, establish that SAE clamping is uniquely vulnerable, or rank defense families. The comparison equalizes suppression of harmful prompts rather than general capability on benign inputs. We did not measure whether the additive defenses achieve their robustness through greater degradation of benign behavior.

F.5 Geometry of the recovery direction

We compare the optimized Δ with the clamped features' neighborhood and reference sets as a descriptive geometry diagnostic. We measure the mean absolute cosine between Δ and three reference sets: the non-orthogonal neighborhood of the clamped features, unrelated SAE features, and random directions. The values are 0.0258, 0.0174, and 0.0164 respectively, so alignment with the real neighborhood exceeds unrelated features by $1.48\times$ and random directions by $1.58\times$. The random baseline $1/\sqrt{d}$ is 0.021. Separately, the clamped feature has near-duplicates in the dictionary at layers 18 and 19, with maximum $|\cos|$ of 0.94 and 0.96.

This is an unsigned alignment diagnostic: it does not distinguish parallel from antiparallel directions and therefore cannot identify a substitutable JumpReLU feature. The effect is modest, and we treat it only as a correlate. The signed width-matched analysis in Appendix F.8 addresses the dictionary-level version of that distinction.

F.6 Dictionary geometry on a second model

This subsection reports an unsigned redundancy diagnostic rather than identifying an underlying behavioral computation. For each layer we take the decoder matrix and compute, for every feature, the largest $|\cos|$ to any *other* feature in the same dictionary. Table 17 reports the distribution for Gemma Scope 2B (width 65k, the dictionary used throughout the paper) and Gemma Scope 9B (width 16k), against a matched control of random unit vectors at the same shape.

Table 17: Largest unsigned cosine from each decoder direction to another direction in the same dictionary. Trained dictionaries contain strongly collinear pairs that matched random dictionaries do not. Because sign is discarded, this table measures redundancy rather than JumpReLU substitutability. Table 19 gives the signed analysis.

Dictionary	Layer	median	p90	max	> 0.7	> 0.9
Gemma Scope 2B width 65k, d=2304	8	0.359	0.617	0.996	6.40%	0.34%
	12	0.342	0.553	0.995	3.41%	0.35%
	16	0.366	0.585	0.995	4.30%	0.23%
	18	0.381	0.631	0.993	6.41%	0.21%
	19	0.371	0.596	0.994	4.53%	0.18%
Gemma Scope 9B width 16k, d=3584	13	0.257	0.439	0.995	1.46%	0.38%
	20	0.287	0.469	0.995	1.66%	0.34%
	26	0.303	0.462	0.983	1.23%	0.33%
	30	0.292	0.470	0.990	1.66%	0.49%
	31	0.292	0.467	0.987	1.24%	0.37%
Random, 65536×2304	—	0.092	0.100	0.139	0.00%	0.00%
Random, 16384×3584	—	0.068	0.075	0.095	0.00%	0.00%

What this does and does not establish. It establishes that strong unsigned decoder redundancy appears in both trained dictionaries and not in the matched random controls: random dictionaries produce no neighbour above 0.7, whereas both trained dictionaries reach $|\cos|$ above 0.98. This geometric result alone does not establish behavioral recovery. The separate 9B recovery experiment is reported in Appendix F.7. The comparison across the two Gemma model sizes is also not exactly matched: the two dictionaries differ in width, residual dimension, and sparsity level, so the absolute medians are not directly comparable across models. The column that carries the argument is the last one, where both models sit in the same range and the random control sits at zero.

F.7 Replication on a second model

Appendix F.6 shows the geometric precondition holds in a second dictionary. This subsection carries the behavioral result across as well, by mining a refusal feature set for `google/gemma-2-9b-it`, recomputing its valid-flip set from scratch, and re-running recovery and judging unchanged.

Mining a 9B feature set. We use the project's own selection pipeline rather than a substitute: a steering vector from a harmful-versus-harmless contrast, integrated-gradient attribution over SAE latents, a per-layer top-k by similarity to that steering vector, and a global top-20. Against the `gemma-scope-9b-pt-res` width-16k dictionaries it places 20 features on layers 21–28. Two model-keyed constants had no 9B entry and we filled them rather than working around them: the steering-vector layer, set to 24 to match the $15/26$ relative depth used for 2B, and the count of leading beginning-of-sequence tokens to skip, set to 1 after checking that both checkpoints tokenize the same chat template to identical ids. The selection is near-identical across the three benchmarks, which is evidence it converged rather than fitting the sample.

Strict harmfulness funnel. We rebuild the valid set from all 520 AdvBench prompts using HarmBench only. The 9B base model produces 14 harmful outputs. The protocol clamp suppresses eight and leaves six harmful before the recovery procedure. Thus clean coverage is $8/14 = 57.1\%$, compared with $29/31 = 93.5\%$ for the 2B intervention. The eight successfully suppressed prompts form the strict recovery set.

Table 18: Strict HarmBench-defined replication on the second checkpoint. The interval is wide because only eight of 14 harmful base outputs are suppressed and therefore eligible for recovery.

Model	Base harmful	Suppressed	Strict n	Recovered harmful	Wilson 95% CI
`gemma-2-2b-it`	31/520	29/31	29	6/29 (20.7%)	[9.9, 38.4]
`gemma-2-9b-it`	14/520	8/14	8	**7/8 (87.5%)**	[52.9, 97.8]

The $7/8$ result shows that classifier-judged recovery under an active clamp is not confined to the 2B checkpoint. It does not support a magnitude or scale comparison: the conditioned sets contain different prompts, use separately mined feature circuits, and the 9B interval is extremely wide. The more immediate 9B failure is clean suppression itself: six of 14 harmful base outputs remain harmful under the clamp before recovery begins.

F.8 Does a wider dictionary close the neighbourhood?

One geometric hypothesis is that correlated decoder directions provide nearby alternatives to a clamped feature. Gemma Scope ships a width ladder at a single layer for both models. For each dictionary we sample 4096 query features and compute the largest *signed* cosine to another decoder direction. Signed cosine is the relevant statistic because JumpReLU coefficients are nonnegative. A large negative cosine is not a substitutable direction. We compare with random unit dictionaries at the same width and residual dimension.

Table 19: Signed nearest-neighbour cosine across dictionary widths. Trained dictionaries have substantially closer positive neighbours than width-matched random dictionaries. This is a geometry diagnostic, not a recovery experiment.

Dictionary	Width	Median	p90	> 0.7	Random med.	Excess
9B, layer 20 $d = 3584$	16k	0.280	0.443	0.61%	0.066	0.214
	32k	0.313	0.487	0.98%	0.068	0.245
	65k	0.339	0.512	1.25%	0.071	0.268
	131k	0.357	0.522	1.61%	0.074	0.284
	262k	0.388	0.542	1.54%	0.076	0.312
2B, layer 12 $d = 2304$	16k	0.322	0.492	1.07%	0.082	0.240
	65k	0.330	0.513	1.90%	0.088	0.242
	131k	0.355	0.539	2.03%	0.092	0.264
	262k	0.375	0.554	2.39%	0.095	0.281

Across widths, trained dictionaries retain excess positive-neighbour similarity beyond the matched random baseline. The excess grows from 0.214 to 0.312 on 9B and from 0.240 to 0.281 on 2B. This shows that widening alone does not remove the geometric correlate. It does *not* show that a wider SAE fails as a behavioral mitigation: answering that question requires reselecting features and rerunning suppression and recovery for every dictionary.

F.9 Preliminary mitigation: constraining the residual channel

Setup. We remove the SAE reconstruction residual from the intervention operator, so the defense constrains the SAE feature coordinates and the residual channel jointly, and re-run the identical recovery procedure on the same $n = 24$ prompts.

Results. Classifier-judged harmfulness falls from $5/24$ to $0/24$. The saved automatic quality audit flags all 24 decoder-only outputs as degenerate, so the intervention blocks recovery by destroying output quality rather than by selectively removing the behavior and is not usable as stated. The lexical detector nevertheless scores all 24 as "non-refusal," another demonstration that it is not a behavioral evaluator. This is the decoder-only setting in which Proposition 1 does not apply, since the reconstruction is lossy.

What a usable mitigation would require. Adversarial training against the post-clamp recovery objective, reported with a utility–robustness trade-off curve on retained capabilities, is the experiment this result motivates. We have not run it and do not claim a working defense.

G Additional Experimental Details

Experimental details. Table 20 summarizes the model, SAE release, intervention target, recovery objective, and evaluator used in each experiment. The supplementary material includes the claim-

Table 20: Experimental configurations. Controlled comparisons hold the recovery machinery fixed where possible, but different defenses and conditioned sets still limit causal comparison.

Experiment	Model	SAE release	Layer(s)	Feature set	Clamp	Loss / target	Projection
TPP	`google/gemma-2-2b`	`gemma-scope-2b-pt-res`	5	official TPP latents	0	target probe logit	encoder
WMDP-Bio	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`, `width_16k/average_l0_59`	3	knowledge features	≤ 0	correct-choice CE	encoder
IOI	GPT-2 Small	`gpt2-small-res-jb`	task layer	IOI-attribution features	0	IO–S logit diff	encoder
Refusal AdvBench	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`, width 65k	8, 10–19	`benchmark_our/global`, 20 feats	3.0	behavior + base-like loss	Jacobian
Refusal HarmBench-Test	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`, width 65k	8, 10–19	`benchmark_our/global`, 20 feats	3.0	behavior + base-like loss	Jacobian
Feature-size sweep	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`, width 65k	11	local-union top-K	3.0	behavior + base-like loss	encoder
Budget diagnostic	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	task layer	matched valid slice	task-specific	task-specific recovery loss	encoder / none
Experiments introduced in this version. All use the atomic hook of Equation 2 and the HarmBench evaluator of Appendix B.2.							
Ordering comparison	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	behavior + base-like loss	Jacobian
Budget sweep (refusal)	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	behavior + base-like loss	Jacobian
Hard-drift audit	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	behavior + base-like loss	Jacobian, no fallback
Shared-direction two-fold CV	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	shared prompt-anchor loss	Jacobian
Matched/deranged target	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	paired target loss	Jacobian
Strict 9B replication	`google/gemma-2-9b-it`	`gemma-scope-9b-pt-res`, width 16k	21–28	mined top-20	3.0	behavior + base-like loss	Jacobian
CAA / mean-shift defense	Gemma-2B	— (no SAE)	steering layer	additive steering vector	—	behavior + base-like loss	—
Residual-constrained mitigation	`google/gemma-2-2b-it`	`gemma-scope-2b-pt-res`	8, 10–19	`benchmark_our/global`	3.0	behavior + base-like loss	Jacobian

audit scripts and sanitized aggregates used to check the manuscript. The full experiment archive will accompany the submission. The Gemma checkpoints come from the Gemma 2 release (Gemma Team, 2024), and the Gemma Scope dictionaries come from the corresponding open SAE suite (Lieberum et al., 2024). The IOI experiment uses GPT-2 Small (Radford et al., 2019). The WMDP-Bio task is drawn from the original WMDP benchmark (Li et al., 2024).

Refusal recovery hyperparameters. The classifier-defined 2B refusal run uses the atomic pre-clamp insertion order, an eight-probe Jacobian sketch with a drift-aligned probe, five correction steps at learning rate 0.05, and relative-ℓ_2 drift weight $\lambda_{\text{act}} = 200$. A 20-step boundary stage uses learning rate 0.08 and norm cap 30. The 80-step recovery stage uses norm cap 40. The saved combined perturbation norms are at most 50 (mean 49.7 over the 29 same-run valid flips, and 50.0 for all six successes). Thus 40 is a stage-2 cap, not a global cap. Drift-constrained checkpoint selection prefers relative defended-feature drift at most 0.03 and otherwise falls back to the best penalized checkpoint. In the primary rerun only $1/29$ same-run valid flips meets 0.03, and none of the six successful recoveries does. The observed drift is therefore reported rather than described as constrained below the threshold. The hard constraint audit disables this fallback and tests $\tau \in \{0.01, 0.03, 0.05, 0.10\}$ at cap 50. Infeasible rows remain denominator failures. The absolute-budget audit tests caps $\{10, 20, 30, 40, 50\}$ with the historical selector. Generation is capped at 128 new tokens. The shared-direction experiment uses the same layer and token support with matched per-layer norm controls. Its re-encoding drift was measured after reconstructing the original directions and checking every stored perturbation norm. The shared and per-prompt rates are not compared numerically because the two procedures optimize different objects.

Defended feature set and encoder family. The refusal experiments clamp 20 SAE features distributed over 11 layers of `google/gemma-2-2b-it` (layers 8 and 10 through 19), taken from the `benchmark_our`/global selection and held at $c_{\mathcal{S}} = 3.0$. The SAEs are Gemma Scope JumpReLU residual-stream autoencoders of width 65k, whose encoder is coordinatewise. This meets the coordinatewise-encoder assumption of Proposition 1 at an individual hook, but the deployed refusal intervention spans multiple layers and therefore does not meet the proposition's single-layer assumption. We do not claim end-to-end order equivalence for this configuration. Its ordering is enforced by the atomic hooks, and re-encoding drift is measured empirically as described in Appendix C.2.

Compute resources. All experiments use frozen language models and frozen SAEs. We train no new language models and no new SAEs. The reported experiments optimize only recovery perturbations, whether per-example, shared across a prompt split, or soft-suffix baselines. Table 21 covers the settings for which we logged runtime. The added experiments of Sections F.1–F.9 reuse the refusal AdvBench configuration and run at comparable cost per condition. We did not instrument them separately and therefore do not report specific runtimes rather than estimate them.

H Limitations

We show recovery under declared access, not an identified surviving computation. This is the sharpest limit on our claims. On the strict 29-prompt set, a deranged target reaches $3/29 = 10.3\%$ harmfulness, statistically indistinguishable from the matched target's $5/29 = 17.2\%$ (exact McNemar $p = 0.6875$). Target-token F1 is higher for matched targets, but it is a diagnostic excluded from

Table 21: Approximate compute resources. Values document scale rather than benchmark runtime.

Experiment	Hardware	Runtime	Runs	Notes
TPP	RTX 6000 Ada-class GPU	several GPU-hours	1 main sweep	official layer-5 benchmark
WMDP-Bio	RTX 6000 Ada-class GPU	$<$ 2 GPU-hours	1 matched strict slice	91 permutations / 6 questions
IOI	RTX 6000 Ada-class GPU	$<$ 1 GPU-hour	1 supporting run	37 valid flips
Refusal AdvBench	2$\times$ RTX 6000 Ada 46GB	$\sim$4 GPU-hours	original comparison	24 lexical-valid prompts
Refusal HarmBench-Test	RTX 6000 Ada-class GPU	appendix run	2 shards merged	43 lexical-valid prompts from 159 total
Feature-set sweep	RTX 6000 Ada-class GPU	$\sim$20 GPU-hours	$K = 1$–61 sweep	full 520-prompt AdvBench slice
Budget diagnostic	RTX 6000 Ada-class GPU	$<$ 1 GPU-hour	small diagnostic	six WMDP valid flips

the success criterion. The results demonstrate recovery under the declared internal-access condition and are compatible with optimized generic residual rewriting. We do not claim to have isolated or restored a pre-existing computation.

The recovery path is localized, not identified. Because the optimizer is projected off the selected encoder directions, mass landing outside the monitored coordinates is expected by construction, and because decoder directions are non-orthogonal, the decomposition is not unique. Our claim is descriptive. The identification experiment, comparing SAE widths, reconstruction qualities, and random dictionaries, requires several dictionaries per layer and a full pipeline re-run per dictionary. We have not run it.

Small evaluated sets, and why they are small. Conditioning on successful suppression makes the refusal sets small. The original 24-prompt set was selected by a lexical detector. The full 520-prompt AdvBench audit finds 30 classifier-defined harmfulness flips under a historical three-layer audit, whereas the exact 11-layer protocol clamp produces 29 flips among 31 harmful base outputs. Fresh generation reproduces the same 29 prompts with $6/29$ fallback-enabled recoveries. Neither set is a population-level sample, and the Wilson intervals remain wide. The shared-direction analysis evaluates the same 29 prompts once each. Its five random seeds are repeated controls, not 145 independent samples. The non-SAE comparison uses the same 29 prompts but remains one conditioned comparison. These experiments establish failure cases and transfer, not calibrated deployment-time rates.

Second model, and what is still missing. The main text reports `google/gemma-2-2b-it`. The circuit-level result additionally uses GPT-2 Small. On `google/gemma-2-9b-it`, a separately mined feature set suppresses only eight of 14 harmful base outputs, and recovery makes seven of those eight harmful (87.5% [52.9, 97.8]). The strict denominator is only eight, and its prompts and circuit differ from the 2B set. This establishes another recovery instance at a second size within the Gemma family but cannot support a magnitude, scale, or cross-family comparison.

The attribution experiment is still incomplete. Identifying what carries recovery requires behavioral reruns across dictionary widths, reconstruction qualities, and matched random dictionaries. Our signed-cosine sweep shows only that trained dictionaries retain excess positive-neighbour similarity across widths. Appendix F.8 reports that geometric diagnostic. It does not test whether widening mitigates recovery. Until the full reruns are done, we make no attribution to an underlying computation beyond the descriptive claim of Section 6.4.

Scope of the transfer results. The shared direction transfers across prompts but is learned per model and per selected feature set. The evidence does not establish transfer beyond that fixed setting. Its re-encoding drift is now measured, but its rate cannot be compared directly with constrained per-prompt optimization because the two procedures optimize different objects. The fixed defense transfers to JailbreakBench but yields no recovery on MaliciousInstruct, so the distribution result is mixed. The extension beyond SAE clamps covers two additive steering defenses. Neither is breached under the matched search, which delimits rather than broadens the SAE-clamp claim. A second model family was not evaluated because the available SAE suite covers only Gemma. Cross-family transfer remains future work.

Internal diagnostic, not a deployment procedure. The procedure assumes access to internal activations and gradients and optimizes residual perturbations directly. It is appropriate for testing

intervention completeness but does not show that an external user can produce the same outcome. An input-only or black-box attack would require a separate method and evaluation.

Feature-selection and release dependence. The tested defenses act on selected SAE features in specific dictionaries and model settings. Different SAE objectives, denser dictionaries, broader multi-layer clamps, or interventions trained explicitly against a post-clamp recovery objective may change the observed trade-offs, and our mitigation result suggests the last of these is the most promising direction.

Legacy metrics in the appendix. Several appendix tables predate the harmfulness audit and report lexical substring and base-like metrics. They are labelled as such and are used only for comparability with prior work and for ruling out alternative explanations, never as evidence of harmfulness.