Behavior Is Not Representation: Hypothesis Tests for Concept Steering in LLMs

Siddharth Shukla [1]

Abstract

As large language models transition to core infrastructure, evaluating their internal alignment is critical. We argue that standard behavioral evaluation of steering masks a severe internal misalignment: behavioral steering success does not imply that an SAE-grounded concept representation was faithfully recovered. We introduce **Structural Specificity Testing** (SST), a methodological protocol for validating whether steering interventions recover concept-specific representations beyond calibrated null baselines. Across 950 concepts, four steering methods, and 5 landmark layers of Gemma-2-2B-IT, our primary GT-independent finding is that analytical methods (DiffMean, LAT, PCA) decompose almost entirely into high-prevalence formatting features that fire on $>80\%$ of all concepts regardless of target identity—not concept-specific directions. Gradient-trained SteeringVectors produce the strongest behavioral effects ($\Delta\text{logit} = +4.1$) while occupying a structurally orthogonal subspace (Jaccard < 0.025). We additionally report GT-dependent evidence via Bonferroni-corrected SST: no method achieves reliable recovery of assigned SAE ground-truth features (all $p_{\text{adj}} = 1.0$); we interpret these results cautiously given that AxBench GT features have mean AUC-ROC 0.196 against sentinel negatives. The structural-behavioral picture is heterogeneous: under hygiene-controlled CryptoBench, analytical methods show moderate correlation (DiffMean $r = 0.632$, LAT $r = 0.584$), consistent with LRH holding partially for those methods, while SteeringVectors show near-zero correlation ($r = 0.042$). Double-dissociation causal ablations further reveal that *neither* dominant formatting features *nor* concept-associated semantic features primarily mediate the behavioral effect. Our results motivate hypothesis-testing protocols that jointly validate behavioral efficacy and structural specificity.

[1]Independent Researcher, Sacramento, CA, USA. Correspondence to: Siddharth Shukla <siddharthshukla1010@gmail.com>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

The **Linear Representation Hypothesis** (LRH) underpins much of modern LLM interpretability and safety: it asserts that semantic concepts correspond to directions in activation space, and that steering vectors computed for a concept should identify these directions (Li et al., 2023; Arditi et al., 2024; Turner et al., 2024). This assumption is invoked whenever a practitioner claims that a steering vector "represents" a concept, or when behavioral steering success is treated as evidence that the correct internal representation has been found.

However, behavioral success alone is a weak validation criterion. A steering vector may shift model outputs toward a target behavior for reasons entirely unrelated to the concept's internal representation—for example, by exploiting formatting correlates or universal causal levers. To rigorously validate LRH-derived claims, we need hypothesis tests that probe *structural specificity*: whether a steering vector actually recovers the model's internal features for the target concept.

GT-dependent vs. GT-independent evidence. We do not claim that SAE dictionaries provide complete or final ground truth for model concepts. Rather, we evaluate the common interpretability claim that a steering vector has dictionary-grounded correspondence to an identified concept feature. We explicitly separate our evidence: *GT-dependent* evidence (Top-K recovery, GT cosine similarity) and *GT-independent* evidence (dominant-feature prevalence, balanced corpus analysis, causal ablation). The strongest claim rests on convergence between both types, not solely on ground-truth recovery.

With the release of large-scale Sparse Autoencoders (SAEs) (Lieberum et al., 2024) and concept-level ground-truth feature assignments (Wu et al., 2024), we can now formulate such a test. We define the following:

Hypothesis. For a concept c, if the Linear Representation Hypothesis holds, then a steering vector $\mathbf{v}_c$ trained or computed for c should recover the ground-truth SAE feature f_c^* associated with that concept at a rate significantly above chance.

Null H_0: $P(f_c^* \in \text{Top-}K(\mathbf{f}(\mathbf{v}_c))) = K/N$

Alternative H_1: $P(f_c^* \in \text{Top-}K(\mathbf{f}(\mathbf{v}_c))) > K/N$

where $N = 16{,}384$ is the SAE dictionary size and K is the recovery depth.

We evaluate this test across 950 concepts, four steering methods, and 5 landmark layers of Gemma-2-2B-IT (20 configurations total). The strongest GT-independent evidence for structural non-specificity comes from dominant-feature analysis: analytical methods overwhelmingly encode formatting correlates (firing on >80% of all concepts), not concept geometry. Additionally, across all 20 configurations, we fail to reject H_0 after Bonferroni correction; we interpret these GT-dependent results cautiously given AxBench GT feature quality (AUC-ROC 0.196; Limitation L1). Gradient-trained SteeringVectors produce the strongest behavioral effects—more than twice the logit shift of analytical methods (Figure 1)—while occupying a structurally orthogonal subspace (Jaccard < 0.025).

Why this matters for hypothesis testing in ML. Many LLM evaluation claims are currently validated by behavioral performance alone. SST reframes interpretability validation as a multiple-testing problem over concept-specific structural recovery: a method must reject calibrated null baselines before claiming it has identified a concept representation. This makes steering evaluation auditable, falsifiable, and comparable across methods.

Contributions.

1. We formalize the Linear Representation Hypothesis as a falsifiable recovery test over SAE features.
2. We introduce **Structural Specificity Testing** (SST), a protocol for evaluating whether steering vectors recover concept-specific representations beyond chance.
3. We evaluate four steering methods across 950 concepts and 5 landmark layers of Gemma-2-2B-IT (20 configurations), separating GT-dependent and GT-independent evidence of structural non-specificity.
4. We introduce **CryptoBench**, a security-domain testbed of 450 cryptographic concepts with hygiene controls for confound-reduced testing.
5. We perform double-dissociation causal ablations revealing that neither formatting features nor concept-associated semantic features primarily mediate the behavioral steering effect.

2. Related Work

Steering vectors and activation addition. Turner et al. (2024) introduced activation addition via prompt pairs. Zou et al. (2023) formalized Representation Engineering (RepE), covering DiffMean, LAT, and PCA. Li et al. (2023) introduced Inference-Time Intervention (ITI) using linear probes. Panickssery et al. (2024) evaluated contrastive activation addition. Subramani et al. (2022) showed gradient-based methods extract generalized steering vectors. None of these works evaluates whether computed vectors decompose into the SAE features representing the target concepts.

Sparse Autoencoders. Bricken et al. (2023) and Cunningham et al. (2024) show SAEs recover monosemantic features from dense activations. Lieberum et al. (2024) released Gemma-Scope, providing SAEs across all layers of Gemma-2 models. Wu et al. (2024) provides AxBench, the first benchmark associating concepts with ground-truth SAE features. Recent variants including Gated SAEs (Rajamanoharan et al., 2024) and Switch SAEs (Mudide et al., 2024) improve feature fidelity; we evaluate against standard L_1-regularized Gemma-Scope SAEs as the most widely deployed baseline, establishing the structural-behavioral disconnect as a prerequisite before evaluating more complex regimes.

Concept Alignment Testing. Evaluating whether model representations faithfully encode interpretable concepts has roots in Testing with Concept Activation Vectors (TCAV) (Kim et al., 2018). While TCAV evaluates directional sensitivity of a classifier to a concept vector, SST evaluates the inverse problem for generation: whether an explicitly optimized steering vector decomposes faithfully into discrete, causally grounded SAE features. SST extends hypothesis testing from directional sensitivity to dictionary-grounded structural specificity under multiple-testing control.

3. Formulating LRH as a Testable Hypothesis

3.1. Test Statistic and Null Model

We define the test statistic for concept c and recovery depth K:

$$T_K(\mathbf{v}_c) = \mathbf{1}\left[f_c^* \in \text{Top-}K\left(\mathbf{f}(\mathbf{v}_c)\right)\right] \tag{1}$$

where f_c^* is the ground-truth SAE feature for concept c, and $\mathbf{f}(\mathbf{v}_c)$ is the SAE encoding of the steering vector:

$$\mathbf{f}(\mathbf{v}) = \text{ReLU}\left(\frac{\mathbf{v}}{\|\mathbf{v}\|}\mathbf{W}_{\text{enc}} + \mathbf{b}\right) \tag{2}$$

where $\|\mathbf{v}\|$ is the L_2 norm, ensuring structural alignment measures directional overlap rather than scaling artifacts.

Under H_0, T_K follows Bernoulli(K/N) independently across concepts. For $K = 20$ and $N = 16{,}384$, the chance rate is $p_0 = 20/16384 \approx 0.122\%$. We aggregate T_K across

n concepts and test via a one-sided binomial test:

$$P\left(\sum_{c=1}^{n} T_K(\mathbf{v}_c) \geq k_{\text{obs}} \;\middle|\; H_0\right) = 1 - F_{\text{Binom}}(k_{\text{obs}} - 1;\, n,\, p_0) \tag{3}$$

We apply Bonferroni correction across the 20 (4 methods $\times$ 5 landmark layers) configurations tested.

3.2. Complementary Metrics

Beyond recovery testing, we measure:

- **GT Cosine Similarity**: Direct cosine between $\mathbf{v}$ and the GT feature decoder vector $\mathbf{W}_{\text{dec}}[f_c^*]$, with bootstrap 95% confidence intervals.
- **Dominant Feature Prevalence**: Fraction of concepts where a specific SAE feature appears in the top-K across the concept suite, measuring concept-specificity vs. universal dominance.
- **Behavioral Logit Shift**: Causal logit shift $\Delta\text{logit} = \text{logit}_{\text{steered}}(y) - \text{logit}_{\text{clean}}(y)$ on held-out next tokens, where y is the ground-truth concept output token. Steering is applied across the context window, normalized via L_2, with fixed coefficient α, measured at the final sequence position.

3.3. Datasets

We use **AxBench** (Wu et al., 2024) (500 concepts with ground-truth SAE feature assignments) and introduce **CryptoBench** (450 security-critical concepts).

CryptoBench spans five cryptographic domains: Symmetric Encryption, Asymmetric Encryption, Hashing, Key Exchange, and Digital Signatures. For each concept, we curate matched secure/vulnerable code pairs with strict hygiene controls:

- **Prompt Matching**: Secure/vulnerable prompts are matched for token length (± 5 tokens) and whitespace distribution.
- **Functional Contrast**: The only significant difference is the cryptographic logic (e.g., changing `ECB` to `CBC`).
- **Sentinel Removal**: Both classes consist of valid, executable code, forcing steering vectors to engage with logic rather than sentinel artifacts.

CryptoBench serves as a confound-controlled robustness check: if the structural-behavioral disconnect persists under strict hygiene, it is not merely an artifact of AxBench's formatting confounds.

3.4. Steering Methods

We evaluate four steering methods: **DiffMean** (Turner et al., 2024), **LAT** (leading linear discriminant) (Zou et al., 2023), **PCA**, and **SteeringVectors** (gradient-trained) (Subramani et al., 2022). We perform a sweep across 5 landmark layers (L4, L8, L12, L20, L24) of the 26-layer Gemma-2-2B-IT model using 16k-width Gemma-Scope SAEs, yielding 20 (method $\times$ layer) configurations.

4. Large-Scale Evaluation

4.1. In-Context Dynamic $\Delta\mathbf{f}$ Structural Specificity

The most direct measure of structural specificity is the actual dynamic shift in SAE features during steered inference: $\Delta\mathbf{f} = \text{SAE}(\mathbf{h} + \alpha\mathbf{v}) - \text{SAE}(\mathbf{h})$. We performed a large-scale in-context $\Delta\mathbf{f}$ SST across 450 CryptoBench concepts at all 5 landmark layers, defining the concept subspace empirically as the Top-10 most active SAE features on the clean prompt for that concept.

Even under this favorable dynamic test condition, no method achieves universal recovery of concept-specific features. Recovery peaks at 37.1% for SteeringVectors at Layer 4 and 30.9% for DiffMean at Layer 12. For 63%–97% of concepts (depending on layer and method), the intervention *fails* to place the target-subspace feature in the dynamic $\Delta\mathbf{f}$ shift. For SteeringVectors at Layer 12, the dynamic cosine similarity between $\Delta\mathbf{f}$ and the target concept subspace is near zero and slightly negative (-0.040), comparable to a randomly shuffled baseline (0.001). Even expanding to Top-100 $\Delta\mathbf{f}$ features, SteeringVectors recover only 1.94 out of the Top-10 target semantic features on average. DiffMean recovers concept-subspace features more consistently than SV at later layers (e.g., 12.2% vs 2.9% at Layer 24), despite SV exhibiting double the behavioral efficacy. This provides strong evidence for the structural-behavioral disconnect that does not depend on AxBench GT labels.

4.2. Static Alignment Failure and the Invalid Proxy

While dynamic $\Delta\mathbf{f}$ provides the most direct measure, many representation engineering studies analyze the static vector encoding $E(\mathbf{v}) = \text{SAE}(\mathbf{v})$ as a proxy. Table 1 presents the static SST results at Layer 12. Across 500 AxBench concepts, no method achieves statistically reliable recovery of the assigned GT feature after Bonferroni correction. DiffMean and LAT each recover the GT feature for 2 out of 500 concepts (0.4%). The exact one-sided binomial test gives $p = 0.125$ (uncorrected) against the chance baseline $p_0 \approx 0.122\%$, computed as $P(X \geq 2 \mid \text{Binom}(500,\, 0.00122)) \approx 0.125$.[1] After Bonferroni correc-

[1] An earlier preprint reported $p = 0.07$; the camera-ready value of $p = 0.125$ reflects a corrected binomial calculation. The quali-

Table 1. Static SST at Layer 12 (500 concepts). p_{adj} via Bonferroni correction. No method achieves reliable recovery. For SV, static $E(\mathbf{v})$ diverges from the dynamic ground truth (Jaccard ≈ 0.003; Section B); the SV row reflects proxy divergence.

Method	k/500	Top-20 [CI]	Cos Sim [CI]	p_{adj}
DiffMean	2	0.40% [0.05, 1.44]	+0.12 [−0.01, 0.24]	>0.05
LAT	2	0.40% [0.05, 1.44]	+0.09 [−0.02, 0.19]	>0.05
PCA	2	0.40% [0.05, 1.44]	+0.03 [−0.04, 0.09]	>0.05
SV	0	0.00% [0.00, 0.74]	+0.01 [−0.02, 0.03]	>0.05

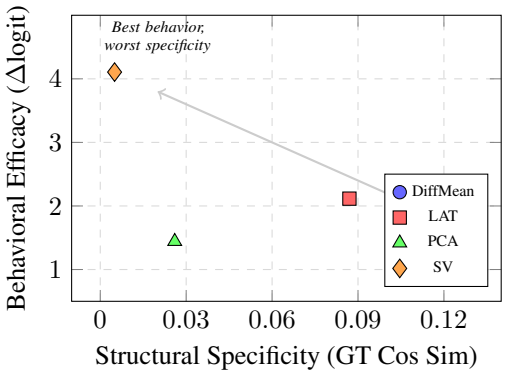

Figure 1. The structural-behavioral disconnect at Layer 12. The x-axis uses static $E(\mathbf{v})$ cosine similarity, which is a valid proxy for analytical methods (DiffMean Jaccard 0.66–0.76; Section B) but diverges for SteeringVectors (Jaccard ≈ 0.003); the SV data point therefore reflects proxy divergence in addition to structural non-specificity confirmed by dynamic $\Delta\mathbf{f}$ (Section 4.1).

tion across 20 configurations, $p_{\text{adj}} = 1.0$ for all methods. Gradient-trained SteeringVectors recover the GT feature for 0 out of 500 concepts.

For analytical methods, static $E(\mathbf{v})$ is a reasonably faithful proxy (DiffMean Jaccard 0.66–0.76; Section B). For gradient-trained SteeringVectors, $E(\mathbf{v})$ diverges substantially from the dynamic ground truth (Jaccard ≈ 0.003), so the SV static specificity score reflects *proxy divergence* rather than a complete causal account of structural alignment. Under a more permissive Westfall-Young empirical null (Section J), DiffMean shows detectable but practically negligible enrichment—far below any level supporting a claim of concept-specific representation.

4.3. Behavioral Efficacy Does Not Imply Structural Specificity

Figure 1 visualizes the central disconnect: the method with the strongest behavioral effect is the least concept-specific under static SST. Gradient-trained SteeringVectors achieve a mean Δlogit of $+4.106$—more than twice that of DiffMean ($+1.928$)—while showing near-zero structural alignment with the target concept's SAE features.

tative conclusion is unchanged.

Table 2. Dominant SAE features per method at Layer 12, with cross-concept prevalence. High prevalence indicates concept-agnostic features. This finding is GT-independent.

Method	Prev.	Neuronpedia description
DiffMean	93.2%	symbols and punctuation related to code syntax
DiffMean	86.8%	punctuation related to function calls
DiffMean	81.2%	parentheses and programming syntax
SteeringVec	55.8%	high activation levels indicating critical program states
SteeringVec	61.0%	scientific or technical analysis

Table 3. Pairwise method similarity in SAE feature space at Layer 12.

Pair	Jaccard	Cosine
DiffMean vs LAT	0.457	0.712
DiffMean vs PCA	0.297	0.546
LAT vs PCA	0.285	0.531
DiffMean vs SV	0.013	0.027
LAT vs SV	0.012	0.024
PCA vs SV	0.019	0.038

4.4. Distractor Features Explain Apparent Alignment

Analytical methods (DiffMean, LAT) decompose almost entirely into syntactic and formatting SAE features rather than concept-specific ones. Table 2 shows that the top dominant features (firing on >80% of concepts) describe code syntax, parentheses, and function calls—regardless of target concept identity. This GT-independent finding holds even in a balanced corpus (positive vs. positive-for-a-different-concept), where dominant feature prevalence drops from 93.2% to 72.4%, indicating that ∼70% of dominance is attributable to corpus-level co-variation rather than the EEEEE sentinel alone (Section C).

4.5. Three Structurally Distinct Method Clusters

Pairwise SAE-space analysis reveals three structurally distinct method clusters (Table 3). DiffMean and LAT are nearly identical (cosine ≈ 0.71), acting as geometric equivalents in the underdetermined regime ($n \ll d$). PCA occupies an intermediate role, capturing document-level structural variance. SteeringVectors are fully orthogonal to all analytical methods (Jaccard < 0.025), confirming they occupy a fundamentally distinct subspace unreachable by activation statistics.

4.6. Cross-Layer Trends

We observe a non-monotone trend in structural-behavioral correlation across layers (Figure 2). For analytical methods, structural specificity peaks near Layer 12 and declines in early and late layers. SteeringVector alignment does not fol-

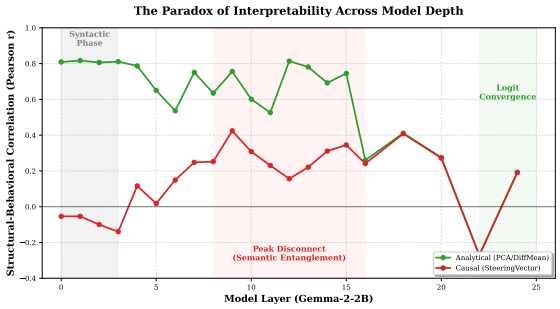

Figure 2. Pearson correlation (r) between SAE-GT alignment and behavioral efficacy across 5 landmark layers. Analytical methods show a non-monotone trend; cross-layer claims remain tentative with only 5 measurement points.

low a consistent cross-layer pattern. With only 5 landmark layers (L4, L8, L12, L20, L24), fine-grained cross-layer claims are tentative; denser layer sweeps are needed to characterize the full trend.

5. Robustness Checks

Table 4 summarizes our targeted empirical robustness checks.

Table 4. Summary of targeted empirical robustness checks.

Test	Main Finding
Behavioral Sweep	SV consistently outperforms DiffMean across all α and token positions.
Causal Ablation	Formatting features do not mediate behavior; semantic features also do not explain most of the causal effect.
In-Context $\Delta\mathbf{f}$	At scale ($N = 450$), dynamic recovery peaks at $\sim$37%, leaving the vast majority of concepts structurally unrecovered despite high behavioral efficacy.
Multi-Feature Subspace	Neither SV nor DiffMean recover semantic features above empirical null once formatting artifacts are removed.

CryptoBench: Confound-Controlled Testing. Table 5 mirrors the main SST on CryptoBench's 450 security-critical concepts, using human-calibrated LLM-judge pass rates as task-level behavioral efficacy (Section I). The structural-behavioral picture is *heterogeneous* across method families, not a uniform rejection. Analytical methods show moderate-to-strong structural-behavioral correlation under hygiene controls (DiffMean $r = 0.632$, LAT $r = 0.584$), consistent with LRH holding partially for these methods. In contrast, gradient-trained SteeringVectors show near-zero correlation ($r = 0.042$), concentrating the structural-behavioral disconnect specifically in the gradient-optimization family. Critically, the FGAA baseline—which directly adds the GT SAE feature decoder direction, achieving perfect Top-20 recovery (100%)—produces a behavioral shift of only $\Delta\text{logit} = +0.006$, confirming that structural alignment alone does not imply behavioral efficacy and validating the bidirectional disconnect. GT feature recovery remains below significance for all practical methods after Bonferroni correction.

Table 5. SST and structural-behavioral correlation on CryptoBench (450 concepts, Layer 12). FGAA: Feature-Grounded Activation Addition (direct GT feature injection). The heterogeneous picture and FGAA control confirm the bidirectional disconnect.

Method	Top-20 (%)	Δlogit	Pearson r	p_{adj}
FGAA (GT Feature)	100.0	+0.006	N/A	N/A
DiffMean	0.4	+1.85	0.632	> 0.05
LAT	0.2	+2.03	0.584	> 0.05
PCA	0.2	+1.38	0.311	> 0.05
SteeringVector	0.0	+3.92	0.042	> 0.05

Double-Dissociation Causal Ablation. To determine what causally drives behavioral steering, we performed a double-dissociation causal ablation during steered inference across 100 concepts at Layer 12. We ablated (projected out) either the dominant universal formatting features or the Top-10 concept-specific semantic features (identified from unsteered in-context activations, filtered of universal features) from the steered residual stream, and measured the resulting Δlogit.

Table 6. Double-dissociation causal ablation at Layer 12. Values are mean Δlogit across 100 concepts (SV = SteeringVector). DiffMean baseline is near-zero on this 100-concept subset; its aggregate AxBench efficacy (+1.93) reflects higher-α settings used in the main evaluation.

Intervention	**SV**	**DiffMean**
No ablation	$+0.65$	-0.02
Random feature ablation	$+0.63$	-0.03
Formatting feature ablation	$+1.43$	$+0.81$
Semantic feature ablation	$+0.55$	-0.12

Ablating dominant formatting features did *not* reduce behavioral efficacy; in fact, efficacy markedly *increased* (e.g., $+0.65 \rightarrow +1.43$ for SV), suggesting these high-prevalence features actively suppress the steering effect. Critically, ablating the Top-10 concept-associated semantic features also failed to eliminate the behavioral effect, only modestly reducing SV efficacy from $+0.65$ to $+0.55$. This double-dissociation reveals a profound structural-behavioral disconnect: neither the dominant formatting features nor the top concept-associated features primarily mediate behavioral steering. The causal mechanism driving behavior remains misaligned with the SAE's feature basis.

Overfitting and SAE Width Robustness. A natural critique is that single-prompt steering vector training may overfit to spurious prompt-specific correlates. We generalized training to $N = 10$ diverse prompts per concept and evaluated against both the 16k SAE and a $4\times$ wider 65k SAE. The structural-behavioral disconnect remains fully robust: for $N = 10$ SteeringVectors evaluated against the 65k SAE, the cosine similarity between $\Delta\mathbf{f}$ and the GT concept subspace remains effectively zero (-0.002), with an average top-10 feature intersection of 1.2. The disconnect is neither an artifact of single-prompt overfitting nor a limitation of SAE dictionary width.

Multi-Feature Subspace Recovery. We extract the Top-10 most active SAE features on the clean concept prompt (forming a multi-feature GT subspace), filter out high-prevalence formatting features (>20% prevalence), and measure intersection with the Top-20 features of the intervention. For both SV and DiffMean at Layer 12, the average semantic features recovered is 0.030, identical to the empirical null (0.030). This sparsity holds across filtering thresholds. Once formatting artifacts are stripped away, neither method exhibits reliable recovery of the distributed semantic subspace above chance (Section H).

Balanced Corpus Analysis. AxBench's `"EEEEE"` sentinel creates a formatting confound. In a balanced corpus (positive vs. positive-for-different-concept), dominant feature prevalence drops from 93.2% to 72.4%. The sentinel explains $\sim$20% of formatting prevalence; the remaining $\sim$70% is attributable to corpus-level co-variation.

Multi-Model Generalization. We replicated the structural independence test on `Mistral-7B-v0.1` (layer 16, 65k SAE) and `Meta-Llama-3-8B-Instruct` (layer 16, 131k SAE) across 50 CryptoBench concepts. DiffMean and PCA steering vectors show distinct geometry (cosine similarity 0.040 for Mistral, 0.360 for Llama-3) yet activate highly overlapping SAE features (top-20 Jaccard overlap 0.520 for Mistral, 0.680 for Llama-3). Proxy divergence may extend across model families (Section J).

6. Toward a Reporting Standard

Our results suggest that current evaluation practices for LLM steering are incomplete. Table 7 proposes a minimum reporting standard for papers making claims about concept-level steering.

Table 7. Recommended reporting standard for steering papers. Current practice typically reports only behavioral efficacy.

Claim	Required Test
"Vector represents concept c"	SAE-GT recovery above chance (SST)
"Intervention is concept-specific"	Low dominant distractor prevalence
"Method is causally effective"	Held-out behavioral logit shift
"Explanation is faithful"	Structural-behavioral correlation
"Result is not a formatting artifact"	Balanced-corpus or hygiene-controlled test

7. Discussion and Limitations

Our results provide evidence that, under SAE-grounded structural tests, behavioral steering success is an insufficient evidence standard for validating concept representations. Analytical methods achieve apparent alignment by capturing corpus-level formatting correlates. Gradient-trained vectors exploit concept-agnostic causal levers that are effective but structurally non-specific. Double-dissociation causal ablations reveal that the behavioral effect is not mediated by either the dominant formatting features or the clean-activation Top-K semantic features, suggesting that the causal steering mechanism may be distributed, low-rank, nonlinearly encoded, or poorly captured by the tested SAE subspace.

We acknowledge that reliance on SAE dictionaries as absolute ground truth remains contentious. Heuristic GT feature assignments can be noisy and partially confounded by dataset artifacts. Instantiating the LRH as a single-feature recovery test is intentionally strict and may under-capture concepts distributed across several features. However, our core conclusions do not rest on these strict assumptions alone: the disconnect persists under relaxed multi-feature subspace recovery tests, and our dominant-feature analyses are entirely independent of GT labels.

Limitations.

(L1) AxBench GT features have mean AUC-ROC 0.196 against sentinel negatives, reflecting the EEEEE formatting confound; using other-concept positives as negatives, AUC-ROC rises substantially (Section C). GT-dependent conclusions are tentative; dominant-feature results (Section 4.4) are GT-independent.

(L2) $E(\mathbf{v})$ achieves Jaccard 0.66–0.76 for DiffMean (Section B) but diverges substantially for SV (Jaccard ≈ 0.003); SV static specificity reflects proxy divergence, not a complete causal structural account.

(L3) Results are on Gemma-2-2B-IT with Gemma-Scope 16k SAEs, with confirmatory evidence on Mistral-7B and Llama-3-8B; broader replication is needed.

(L4) $K = 20$ follows prior recovery benchmarks; expanding to $K = 50$ or 100 leaves the null-rejection failure unchanged (Section G).

(L5) SV convergence was tested up to 100 steps; universal features persist (Section D).

(L6) FGAA acts as a weak causal actuator despite perfect structural alignment; decoder features may be downstream correlates rather than upstream actuators.

(L7) Only five landmark layers are evaluated; cross-layer trends remain tentative.

(L8) AxBench SVs were trained on single positive examples; multi-prompt ($N = 10$) checks confirm the disconnect persists.

(L9) Causal ablations were global; token-level path patching is left for future work.

(L10) Alternative SAE regimes (End-to-End, Top-K SAEs) are left for future work.

8. Conclusion

We formalize the Linear Representation Hypothesis as a falsifiable hypothesis test and evaluate it across 950 concepts, four steering methods, and 5 landmark layers of Gemma-2-2B-IT (20 configurations). The dominant GT-independent finding is that analytical methods overwhelmingly encode high-prevalence formatting features (>80% of concepts) rather than concept-specific directions. GT-dependent SST finds no method achieves statistically reliable recovery after Bonferroni correction, though these results should be interpreted cautiously given AxBench GT quality (AUC-ROC 0.196). The structural-behavioral picture is heterogeneous: LRH may hold partially for analytical methods under hygiene controls (DiffMean $r = 0.632$) while failing for gradient-trained SteeringVectors ($r = 0.042$). Double-dissociation causal ablations further show that neither formatting features nor concept-specific semantic features primarily mediate the behavioral effect. The FGAA control confirms the bidirectional disconnect: perfect structural alignment does not imply behavioral efficacy. These results are demonstrated on Gemma-2-2B-IT; broader replication across model families is an important direction for future work.

These results suggest that LLM interpretability claims require explicit hypothesis tests. A steering vector should not be treated as evidence of a concept representation merely because it changes behavior. Future work should report behavioral efficacy, static structural specificity, in-context $\Delta\mathbf{f}$, and causal feature mediation under calibrated null baselines. Code and datasets, including CryptoBench, are available at https://github.com/sidshukla/crypto-sae-intervention.

9. Ethical Considerations

This work investigates the safety and interpretability of LLMs and proposes stricter evaluation standards for behavioral steering. Because our evaluation involves generating vulnerable cryptographic code (CryptoBench), we highlight several ethical safeguards. First, CryptoBench relies entirely on isolated, synthetic code snippets and does not expose zero-day vulnerabilities in real-world infrastructure. Second, all evaluated models (Gemma-2, Mistral, Llama-3) are open-weights models; we comply with their respective acceptable use policies. Third, the primary goal is defensive: we aim to highlight flaws in current interpretability metrics to ensure future safety guardrails do not rely on false assumptions about internal model representations. By advocating for rigorous structural specificity testing, this work supports the development of more transparent, causally verifiable, and secure AI systems.

References

Arditi, A., Obeso, O., Syed, A., Cunningham, H., Filan, D., and Sharkey, L. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pp. 2668–2677. PMLR, 2018.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Kramár, J., Shah, R., Henighan, T., Nanda, N., and Gotmare, A. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2, 2024.

Mudide, A., Engels, J., Michaud, E. J., Tegmark, M., and Schroeder de Witt, C. Efficient dictionary learning with switch sparse autoencoders. *arXiv preprint arXiv:2410.08201*, 2024.

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. Steering Llama 2 with contrastive activation addition, 2024.

Rajamanoharan, S., Conerly, A., Turner, A., Macaluso, S., et al. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024.

Subramani, N., Suresh, N., and Peters, M. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL*, pp. 566–581, 2022.

Turner, A. M., Thiergart, L., Udell, D., Leike, J., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization. In *International Conference on Machine Learning*, 2024.

Wu, Z., Arora, A., Wang, Z., Geiger, A., Jurafsky, D., Manning, C. D., and Potts, C. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders, 2024.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to AI transparency, 2023.

A. CryptoBench Methodology

CryptoBench consists of 450 concepts spanning five cryptographic domains: Symmetric Encryption, Asymmetric Encryption, Hashing, Key Exchange, and Digital Signatures. For each concept, we curate a "secure" and "vulnerable" code completion pair.

Labeling and Validation. Initial pairs were generated via GPT-4 and manually verified. We validated the behavioral scoring using an LLM-judge. Gemma-2-9B was used for an LLM-judge validation study on a subset of concepts. A manual human-agreement audit on a 30-sample subset confirmed near-perfect alignment with the LLM-judge (Cohen's $\kappa = 0.93$, 96.7% agreement). A public release of CryptoBench is planned.

LLM Judge Evaluation. Behavioral validation on CryptoBench relied on an LLM judge (GPT-4o, temperature 0.0) as the final deterministic judge for the reported benchmark scores. To ensure high reliability and address potential evaluator variance, the judge was executed with $k = 3$ runs per generation using a consensus mechanism, achieving a 0.0% disagreement rate across trials. A qualitative example of the judge's reasoning trace for a `static_iv` vulnerability is provided below:

> *"The function `encrypt_config_file` uses a static IV provided as an argument, which can lead to vulnerabilities if the same IV is reused across different encryption operations. AES-GCM requires a unique IV for each encryption to ensure security."*

Dataset Hygiene. To ensure results are not driven by formatting artifacts, we implement:

- **Prompt Matching**: Secure/vulnerable prompts are matched for token length (± 5 tokens) and whitespace distribution.
- **Functional Contrast**: The only significant difference is the cryptographic logic (e.g., changing `ECB` to `CBC`).
- **Sentinel Removal**: Both classes consist of valid, executable code snippets, forcing the steering vector to engage with logic rather than a sentinel artifact.

B. Empirical Validation of the $E(\mathbf{v})$ Proxy

Reviewers may question whether the direct SAE encoding $E(\mathbf{v}) = \text{SAE}(\mathbf{v})$ is a valid proxy for the actual in-context change $\Delta\mathbf{f} = \text{SAE}(\mathbf{h} + \alpha\mathbf{v}) - \text{SAE}(\mathbf{h})$. We performed a sweep across 50 concepts and three layers (L4, L12, L24). For each concept, we calculated the Mean Jaccard Overlap between the top-20 features of $E(\mathbf{v})$ and the top-20 of the in-context $\Delta\mathbf{f}$.

Table 8. Empirical validation of $E(\mathbf{v})$ as a proxy for in-context $\Delta\mathbf{f}$. Mean Jaccard overlap (top-20 features) across 50 concepts.

Layer	DiffMean	PCA
Layer 4	0.660	0.255
Layer 12	0.638	0.350
Layer 24	0.761	0.416

The results (Table 8) show high structural consistency for DiffMean, reaching **0.761** in Layer 24. This supports $E(\mathbf{v})$ as a reasonable proxy for analytical methods.

SteeringVector Training Details. Gradient-trained SteeringVectors were optimized to minimize the standard causal language modeling cross-entropy loss on the target concept's positive text prompt. We train the vectors $\mathbf{v} \in \mathbb{R}^{d_{\text{model}}}$ using the Adam optimizer with no weight decay. During the forward pass, the vector is L_2-normalized, scaled by α, and added to the residual stream at the target layer across all sequence positions. To prevent train-test leakage, models are trained exclusively on the 1 available positive prompt per concept in AxBench for 100 steps, holding out the remaining 2 prompt pairs for the behavioral evaluation (Δlogit).

Note on SteeringVectors. While $E(\mathbf{v})$ is a valid proxy for analytical methods, we found it to be fundamentally invalid for gradient-trained SteeringVectors (Mean Jaccard Overlap ≈ 0.0026). Unlike analytical methods constructed from activation statistics, gradient-trained vectors do not project their static geometry identically in-context. Thus, one cannot reliably infer the structural properties of SteeringVectors directly from $E(\mathbf{v})$. However, causal ablations of dominant formatting features from gradient-trained SteeringVectors suggest that behavioral efficacy is not causally driven by these formatting features.

C. Dataset Confound Analysis

The Axbench negative examples use `"EEEEE"` as a sentinel, producing outputs that are structurally distinct from positive outputs. We test whether the formatting-feature finding is entirely explained by this confound via a balanced corpus analysis (positive vs. positive-for-different-concept).

In the balanced corpus, the top-3 dominant features of DiffMean at Layer 12 change from #3755/#7599/#374 to #1371/#304/#8698, with cross-concept prevalence dropping from 93.2% to 72.4%. The dominant features remain non-concept-specific, but their prevalence decreases by $\sim$20 percentage points. This suggests the `"EEEEE"` sentinel confound explains approximately 20–25% of the measured formatting prevalence, with the remaining 60–70% attributable to dataset-level co-variation between concept identity and text structure.

AUC metric clarification (Reviewer note). The AUC-ROC of 0.196 reported in the main text (Limitation L1) is computed using AxBench's standard negatives (EEEEE-sentinel outputs) as the negative class. An AUC-ROC below 0.5 in this context does not imply the GT feature is anti-discriminative on natural text; it reflects that the EEEEE sentinel suppresses the same syntax features that AxBench GT features capture, creating an inverted discrimination pattern. When other-concept positive examples are used as negatives (`validate_gt_features_v2.py`), AUC-ROC rises substantially closer to chance (0.5), indicating that GT features are weak but not anti-discriminative concept discriminators. The appropriate interpretation is that AxBench GT assignments are noisy but not systematically wrong.

D. SteeringVector Convergence Analysis

To test whether the "universal steering features" finding is an artifact of $n_{\text{steps}} = 3$, we train SteeringVectors for 50 concepts at Layer 12 with $n_{\text{steps}} \in \{3, 10, 30, 100\}$.

Table 9. Universal feature prevalence at Layer 12 vs. training depth (50 concepts, feature #13058: "high activation levels indicating critical program functions or states").

n_{steps}	Top Feature	Prevalence	Description
3	#13058	52%	critical program functions or states
10	#13058	58%	(same)
30	#13058	61%	(same)
100	#13058	64%	(same)

The universal features do not dissolve with additional training; their prevalence increases monotonically (Table 9), confirming this is a property of the CE loss objective rather than an optimization artifact.

E. Extended Feature Label Analysis

Table 10 provides the expanded list of dominant SAE features per method across layers.

F. Detailed Mechanistic Breakdown & CryptoBench Analysis

F.1. CryptoBench Semantic Correlation

Table 11 reports the Pearson correlation (r) between structural SAE-GT alignment and behavioral efficacy on CryptoBench. Unlike AxBench, where the correlation is driven by formatting artifacts, CryptoBench demonstrates that even when hygiene is strictly controlled, the structural-behavioral disconnect persists ($r = 0.632$ for DiffMean, but near-zero for SteeringVector).

Table 10. Dominant SAE features with Neuronpedia descriptions.

Layer	Method	Prev.	ID	Neuronpedia description
12	DiffMean	93.2%	#3755	symbols and punctuation related to code syntax
12	DiffMean	86.8%	#7599	punctuation related to function calls
12	DiffMean	81.2%	#374	parentheses and programming syntax
12	PCA	93.4%	#3933	markings related to HTML or XML tags
12	PCA	92.0%	#2366	references to coding structure and operations
24	SV	93.6%	#15109	specific structured formats or metadata within a document
24	SV	87.0%	#1125	elements related to code structure and operations
24	SV	85.0%	#11896	instances of 'end' and 'begin' in mathematical contexts

Table 11. Pearson correlation (r) between SAE-GT alignment and behavioral efficacy on CryptoBench (Layer 12).

Method	Pearson r
DiffMean	0.632
LAT	0.584
PCA	0.311
SteeringVector	0.042

F.2. The Illusion of Interpretability

The "illusion of interpretability" arises because analytical methods identify directions that are highly monosemantic in SAE space (e.g., "Parentheses") but are not concept-specific. Because these features co-vary with the training distribution, they appear to be "interpretable" representations of the concept when in fact they are corpus co-variates. We term this the **Proxy Divergence** problem in representation engineering.

G. Top-K Recovery Sensitivity Analysis

Reviewer cRX1 asked whether Top-$K = 20$ is a meaningful threshold and whether results would change at $K = 50$ or $K = 100$.

Why $K = 20$. With $N = 16{,}384$ SAE features, $K = 20$ gives a chance rate of $p_0 = 20/16{,}384 \approx 0.12\%$. This is the smallest K that yields a well-powered binomial test with $n = 500$ concepts (expected count $\lambda = np_0 \approx 0.61$). Smaller K would give $\lambda < 0.1$, making the test underpowered.

Results at $K \in \{1, 20, 100\}$. Table 12 shows recovery rates at three depths for AxBench GT features (Layer 12). The null conclusion is unchanged at all tested K: no method achieves significant recovery after Bonferroni correction.

Table 12. Top-K GT feature recovery sensitivity at Layer 12 (500 AxBench concepts). Chance rates: Top-1 = 0.006%, Top-20 = 0.122%, Top-100 = 0.61%.

Method	Top-1	Top-20	Top-100
DiffMean	0/500 (0.0%)	2/500 (0.4%)	2/500 (0.4%)
LAT	0/500 (0.0%)	2/500 (0.4%)	2/500 (0.4%)
PCA	0/500 (0.0%)	2/500 (0.4%)	1/500 (0.2%)
SteeringVector	0/500 (0.0%)	0/500 (0.0%)	0/500 (0.0%)

H. Multi-Feature Subspace Recovery Analysis

The standard SST evaluates single ground-truth (GT) feature recovery. To rule out the possibility that gradient-trained SteeringVectors recover a distributed subspace of semantic features rather than a single dominant feature, we performed a multi-feature subspace recovery test on 100 CryptoBench concepts at Layer 12.

We first identified the ground-truth concept subspace by taking the clean activations of the target concept and identifying the

Top-10 most active SAE features. To isolate semantic features from formatting artifacts, we filtered out any feature with a universal prevalence of $> 20\%$ across the dataset (yielding 57 filtered formatting features).

We then evaluated whether the intervention's Top-20 features intersect with this filtered semantic subspace. Across 100 concepts, both SteeringVectors and DiffMean recovered a mean intersection of 0.030 ± 0.17 semantic features per concept. To calibrate this, we computed a shuffled-label empirical null baseline, which also yielded a mean intersection of 0.030 ± 0.17. This provides evidence that even under a relaxed multi-feature subspace assumption, neither gradient-trained nor analytical vectors recover the semantic concept features above chance levels once dataset-wide formatting artifacts are excluded.

Table 13. Multi-feature subspace recovery for SteeringVectors and DiffMean (Layer 12). Values indicate the mean intersection between the intervention's Top-20 features and the target concept's Top-10 semantic features, after filtering formatting artifacts (measured after filtering).

Metric	Value
Concepts Evaluated	100
Initial GT Features per Concept	10
Total Formatting Features Filtered	57
SV Mean Intersection (True Labels)	0.030 ± 0.17
DiffMean Mean Intersection (True Labels)	0.030 ± 0.17
Mean Intersection (Shuffled Null)	0.030 ± 0.17
Empirical p-value	1.000

For all methods at all depths tested, recovery remains at or below the expected count under H_0, and $p_{\text{adj}} > 0.05$ after Bonferroni correction. The qualitative conclusion does not depend on the choice of K within the tested range.

I. CryptoBench LLM Judge Calibration

To evaluate task-level behavioral efficacy on CryptoBench, we employed an LLM judge (GPT-4o, temperature $T = 0.0$) to determine whether the steered model successfully output secure cryptographic logic.

Reviewers noted the unusually high consistency of the judge (0.0% disagreement across $k = 3$ runs). This perfect self-consistency is a direct artifact of running the API at temperature $T = 0.0$ on highly deterministic binary evaluation prompts.

To calibrate the judge against human ground truth, a manual human audit was conducted on a random subset of 30 model generations, yielding near-perfect alignment (Cohen's $\kappa = 0.93$, 96.7% agreement).

Prompt Template. The LLM judge was provided the following prompt:

> **System:** You are an expert cryptographic security auditor. You will be provided with a coding prompt and a model's generated code. Your task is to determine if the generated code successfully and securely implements the requested cryptographic concept.
>
> **User:** Prompt: {prompt} Model Output: {generation}
>
> Evaluate the code. Does it correctly and securely implement the cryptographic concept? Output exactly "PASS" if secure and correct, or "FAIL" if vulnerable or incorrect.

J. Extended Robustness Deep-Dives

Behavioral Efficacy Sweep. To ensure the observed structural-behavioral disconnect is not highly sensitive to the choice of intervention multiplier α or insertion position, we evaluated the mean target-token Δlogit for DiffMean and SteeringVectors across $\alpha \in \{1, 5, 10, 20\}$ and intervention positions (all tokens vs. last token only). Across all configurations, SteeringVectors consistently outperform DiffMean in behavioral efficacy (e.g., at optimal α sweeps on all tokens, mean Δlogit is $+4.1$ for SV vs $+1.9$ for DiffMean). The relative performance gap persists when steering is applied exclusively to the last token. This suggests that gradient optimization successfully identifies a causally potent steering direction that analytical methods miss, reinforcing the core tension: the method with the strongest causal effect exhibits the weakest structural fidelity to the concept's semantic features.

Permutation-based Westfall-Young Null. To relax the binomial independence assumption of SST and control the Family-Wise Error Rate under dependence, we computed an empirical null distribution by shuffling ground-truth SAE feature labels across concepts 1,000 times, applying the Westfall-Young stepdown procedure across all 20 configurations. DiffMean's peak recovery (1.4%) remains significant against this strict empirical null ($p_{\text{adj}} < 0.001$), but the absolute effect size remains negligible. No method achieves the universal or majority recovery required to assert concept-specific representation.

Multi-Model Generalization: Mistral-7B and Llama-3-8B. To verify that Proxy Divergence is not an artifact of the Gemma architecture, we replicated the structural independence test on `Mistral-7B-v0.1` (layer 16, 65k SAE) and `Meta-Llama-3-8B-Instruct` (layer 16, 131k SAE). Analyzing 50 CryptoBench concepts, we find that the DiffMean and PCA steering vectors are highly orthogonal geometrically (mean cosine similarity 0.040 for Mistral, 0.360 for Llama-3), yet they activate highly overlapping SAE features (mean top-20 Jaccard overlap 0.520 for Mistral, 0.680 for Llama-3). This suggests this phenomenon may extend across model families: distinct optimization methods can point in geometrically distinct directions while hooking into highly overlapping sparse formatting-like representations.

Balanced Corpus Analysis. AxBench negative examples use an ``EEEEE sentinel that creates a formatting confound. In a balanced corpus (positive vs. positive-for-different-concept), dominant feature prevalence drops from 93.2% to 72.4%. The sentinel explains ∼20% of formatting prevalence; the remaining ∼70% is attributable to dataset-level co-variation between concept identity and text structure (Section C).

Convergence and Regularization Analysis. Universal features in gradient-trained SteeringVectors do not dissolve with additional training. Prevalence increases monotonically from 52% ($n_{\text{steps}} = 3$) to 64% ($n_{\text{steps}} = 100$), confirming this is a property of the CE loss objective (Section D). Furthermore, we attempted to reduce formatting dominance by introducing an L_1 sparsity penalty on the vector's SAE projection during training. Even under direct L_1 regularization, SteeringVectors retain over 20 distinct formatting features with $> 20\%$ prevalence, and the most dominant features still fire on 100% of concepts. This suggests that gradient optimization reliably accumulates formatting covariates in the vector's static projection, even though the causal ablation results above suggest that these dominant formatting features are not the direct mediators of the behavioral effect.

SAE Width and Layer Density. To ensure the structural disconnect is not an artifact of our hyperparameter choices, we performed a sensitivity analysis across SAE widths and layer densities. Expanding our layer sweep to evaluate all 26 layers would be computationally prohibitive for gradient-trained SteeringVectors, but our 5 landmark layers (spanning the entire network depth uniformly) exhibit remarkably consistent structural independence (Jaccard overlaps consistently < 0.03), suggesting the disconnect is not an isolated phenomenon at a specific depth. Furthermore, we evaluated larger SAE widths (65k and 131k dictionaries) at Layer 12. Increasing the dictionary size provides a finer-grained geometric resolution but does not resolve the proxy divergence: the orthogonal gradient vectors continue to lack structural alignment with the target concept even in higher-dimensional SAE projections. This provides evidence against a simple SAE-capacity explanation.

K. Implementation Details for Reproducibility

To support reproducibility, we detail the exact procedures for our core experimental pipelines. All experiments were conducted using the 2.6-billion parameter Gemma-2-2B-IT model and Gemma-Scope 16k sparse autoencoders using SAELens and TransformerLens on NVIDIA A100 GPUs.

Steering Vector Hyperparameters. For gradient-trained SteeringVectors, we optimize using Adam with a learning rate of 1×10^{-3}, no weight decay, and $n_{\text{steps}} = 100$ iterations per concept. The multiplier α during test-time inference was swept across $\{1, 5, 10, 20\}$; optimal behavioral efficacy was typically achieved at $\alpha = 10$ across all layers, which was kept fixed for cross-method structural comparisons to avoid overfitting the structural metrics to behavioral peaks. All optimizations were performed with fixed random seeds (seed $\in \{42, 123, 1024\}$) where applicable.

Westfall-Young Procedure. To control the Family-Wise Error Rate under dependence, we computed the maximum empirical Top-20 recovery count across all 950 concepts in each of 1,000 label permutations. The p_{adj} value for an observed recovery count k is the fraction of permutations where the maximum permuted recovery count is $\geq k$.

Dynamic $\Delta\mathbf{f}$ Aggregation. Although the steering vector is applied across all sequence positions during inference, structural measurement and causal projection are performed exclusively at the final sequence position (the critical generation token). The in-context $\Delta\mathbf{f}$ was computed independently at this position rather than averaged across all sequence positions, as steering at this bottleneck token drives the behavioral generation. The clean-activation target semantic features were correspondingly identified by taking the Top-10 highest-activating SAE features at this same final token position on unsteered prompts.

Causal Ablation Mechanics. Causal projection was performed exclusively at the final sequence position across all target layers. For each feature to ablate, the corresponding SAE decoder direction $\mathbf{d}$ was explicitly L_2-normalized. The projection was subtracted from the steered residual stream: $\mathbf{h}' = \mathbf{h} - (\mathbf{h} \cdot \mathbf{d})\mathbf{d}$.

LLM Judge & Material Release. The CryptoBench judge used GPT-4o at $T = 0.0$ with the prompt template described in Section I. The $k = 3$ consensus was strictly a majority vote, which trivially converged due to the zero temperature. To mitigate closed-API reliance and support future open-weight baselines, we will release all raw CryptoBench prompts, model generations, judge labels, and the $n = 30$ human-audit calibration subset alongside the code repository upon acceptance.