SHADOW ANCHORING: GROUNDING HUMAN-INTERPRETABLE CONCEPTS IN TRANSFORMERS VIA SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability methods for language models are mostly post hoc: probes and sparse autoencoders (SAEs) are fitted after training to discover features the model happened to learn. We study a proactive alternative, concept anchoring, in which designated latent coordinates are trained to carry human-specified concepts, and ask whether it can be retrofitted into a pretrained transformer without degrading language modeling (an "interpretability tax"). We use continuous RGB color in CSS code as a machine-checkable testbed, since hex literals provide exact three-dimensional ground truth inside natural, syntactically constrained text. We introduce SAE shadow anchoring: rather than using a sparse autoencoder as an observational post-hoc probe, we repurpose a frozen, pretrained SAE at an intermediate layer of Qwen2.5-Coder-0.5B as an audited scaffolding dictionary, backpropagating concept supervision through its encoder into LoRA adapters. Compared with anchoring the raw residual stream, shadow anchoring achieves higher continuous color recovery while reducing the language-modeling cost relative to unconstrained fine-tuning (cutting the evaluation-measured tax from 0.26 to 0.10 percentage points). Clamping the anchored latents monotonically shifts target color values in generated CSS with 100% syntactic validity and introduces no spurious color tokens on general Python and JavaScript algorithms. The representation transfers across six syntactic contexts, and Hewitt–Liang control tasks are consistent with continuous geometric representations rather than random target fitting. Separately, digit-pair tokenization lets a 38M-parameter transformer trained from scratch generalize to unseen hex codes.

1 INTRODUCTION

Modern language models are typically treated as black boxes whose internal computations connecting inputs to outputs remain opaque. Two broad paradigms have emerged to address this opacity. The dominant approach is *post hoc*: models are trained without constraint, after which linear probes (Alain & Bengio, 2016), concept activation vectors (Kim et al., 2017), or sparse autoencoders (Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2026) are fitted to discover what features the network learned. A second, proactive approach is to structure representations *during* training to align with human-specified concepts. Concept bottleneck models (Koh et al., 2020) and concept whitening (Chen et al., 2020) established the feasibility of this principle in vision classifiers, and recent concurrent work on sparse concept anchoring (Fraser & Wielopolski, 2025) applies geometric anchor losses under minimal supervision.

However, existing training-time concept structuring has been demonstrated almost exclusively on small, purpose-built architectures—shallow autoencoders or linear bottlenecks lacking self-attention, multi-layer residual streams, or autoregressive language modeling objectives. Directly retrofitting proactive concept structures into the entangled residual stream of an *already-pretrained* foundation model risks degrading language modeling fluency (an "interpretability tax"). In this work, we show this cost can be reduced by repurposing a frozen sparse autoencoder not as an observational post-hoc lens, but as an active scaffolding coordinate system for training-time supervision during parameter-efficient adaptation.

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 investigate this framework using continuous RGB color as a machine-checkable testbed. Color in source code (e.g., CSS hex literals) provides an exact, continuous, three-dimensional ground truth that decomposes deterministically into numeric channels while embedded inside naturalistic, syntactically constrained programming text. This setup allows us to anchor concepts with objective supervision and evaluate continuous recovery rigorously.

Contributions.

- **Pretraining anchoring with atomic tokenization:** We show that structural digit-pair tokenization enables a 38M-parameter transformer trained from scratch to achieve $r = 0.996$ continuous out-of-distribution generalization on synthetic unseen hex codes, overcoming the memorization pitfalls of standard tokenizers.
- **Layer-wise representation geometry:** Linear probing across layers of a 0.5B-parameter code LLM (Qwen2.5-Coder-0.5B) reveals an intermediate semantic plateau spanning Layers 4–12 (peaking at Layer 8, $R^2 \approx 0.89$), motivating Layer 8 for continuous concept intervention.
- **SAE Shadow Anchoring:** We introduce a paradigm that routes supervision through a frozen sparse autoencoder, cutting the downstream language modeling interpretability tax relative to unconstrained fine-tuning from -0.26 ± 0.03 to -0.10 ± 0.02 percentage points under a fixed training seed (an advantage exceeding evaluation data-sampling noise across 5 evaluation splits, paired t-test $p < 0.001$), while establishing an intrinsic coordinate interface with near-perfect continuous recovery ($r = 0.9871 \pm 0.0022$, MAE $= 0.0305 \pm 0.0010$).
- **Causal steering vs. contrastive baselines:** Layer 8 intervention hooks demonstrate that clamping anchored latents causally steers generation monotonically (elevating mean target green from 0.256 to 0.559 at $\alpha = 15$, a $+118.2\%$ relative shift) with 100% syntax validity and zero spurious color tokens on general Python and JavaScript algorithms. While target elevation is substantial, intermediate activation steering induces broader color-subspace co-elevation across channels. Furthermore, sparse latent clamping avoids the severe collinearity ($\cos = 0.608$) and syntactic degradation (80% validity at $\alpha = 15$) of training-free contrastive activation addition (DiffMean/CAA).
- **Syntactic generalization and shortcut diagnostics:** The anchored representation robustly transfers across 6 distinct syntactic paradigms spanning CSS variables, HTML attributes, and JavaScript/Python literals ($r \in [0.71, 0.82]$). Hewitt–Liang control tasks establish large selectivity margins ($+1.0704$ on raw activations, $+1.1610$ across the SAE dictionary), confirming consistency with genuine continuous geometry rather than random target fitting.

2 Related Work

Post-hoc representation analysis and emergence. Probing (Alain & Bengio, 2016) and concept vectors (Kim et al., 2017) show human-interpretable concepts are linearly decodable from intermediate representations (Park et al., 2023). Pure next-token prediction can induce internal world models, from Othello board states (Li et al., 2022; Nanda et al., 2023b) to color spaces (Abdou et al., 2021). Grokking transitions from memorization to generalized circuits (Power et al., 2022; Nanda et al., 2023a), and intermediate layers repeatedly carry peak decodable task representations before output specialization (Tenney et al., 2019; Skean et al., 2025).

Training-time concept structuring. Concept bottlenecks (Koh et al., 2020) and concept whitening (Chen et al., 2020) align latent axes with known concepts in vision models. Sparse concept anchoring (Fraser & Wielopolski, 2025) explored hypersphere regularizers under minimal supervision ($< 0.1\%$) on small autoencoders without attention or residual streams, identifying transformer scaling as open.

Sparse autoencoders and supervised training. Sparse autoencoders resolve superposition by mapping dense activations into overcomplete, sparse feature bases (Elhage et al., 2022; Cunningham et al., 2023; Bricken et al., 2023; Templeton et al., 2026), and can regularize training against forgetting (Morelli et al., 2026). Whereas AlignSAE (Yang et al., 2026) and CB-SAE (Kulkarni et al., 2026) freeze the base LM and adapt the SAE to match concepts, shadow anchoring reverses this: we freeze the SAE as an audited dictionary and backpropagate supervision into the model's adapters.

Representation engineering and causal steering. Inference-time activation addition (Turner et al., 2024; Rimsky et al., 2024) and representation engineering (Zou et al., 2023) derive steering vectors from contrastive activations, and LEACE (Belrose et al., 2023) projects concepts out in closed form. These methods do not alter default representations; our framework modifies designated coordinates during adaptation.

3 METHOD

3.1 CONCEPT ANCHORING OBJECTIVE

Let $h_i \in \mathbb{R}^d$ denote the hidden representation for training token i, and let $c_i \in [0,1]^k$ denote the normalized ground-truth continuous concept vector ($k = 3$ for RGB triples). The anchor loss measures mean squared error over a designated coordinate subset $\mathcal{S}$ at color literal token positions $\mathcal{A}$:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \left\| h_i^{(\mathcal{S})} - c_i \right\|_2^2. \tag{1}$$

To prevent concept leakage into non-target dimensions, we add an L_2 coordinate leakage penalty at color token positions:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{|\mathcal{A}| \cdot |\mathcal{S}^c|} \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{S}^c} (z_i^{(j)})^2, \tag{2}$$

where $z = h$ in Phase 3 and $z = \text{SAE}_{\text{enc}}(h)$ in Phase 4, and $\mathcal{S}^c$ denotes the non-anchored coordinates (893 unanchored hidden dimensions in Phase 3, and 7,165 unanchored SAE latents in Phase 4). The joint training objective combines autoregressive language modeling with anchoring:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}}, \tag{3}$$

with $\lambda_{\text{anchor}} = 5.0$ and $\lambda_{\text{sparse}} = 0.5$. Recovery is diagnosed via Pearson correlation $r = \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y}$ and probe coefficient of determination $R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$.

3.2 BRANCH 1: PRETRAINING ANCHORING WITH ATOMIC TOKENIZATION

In from-scratch pretraining, standard BPE tokenizers often merge frequent hex strings into monolithic tokens, encouraging lookup memorization. We introduce a domain-specific tokenizer that splits hex literals into constituent digit-pair tokens (e.g., `#ff0088` → `["#", "ff", "00", "88"]`). Anchoring is applied at the terminating token of each color literal on coordinates $\mathcal{S} = \{0, 1, 2\}$ of the final hidden state.

3.3 BRANCH 2: LORA-MEDIATED ANCHORING OF PRETRAINED MODELS

To retrofit concepts into a frozen pretrained model (Qwen2.5-Coder-0.5B), we identify the layer with peak decodability via linear probing (Layer 8; Section 5.2.1). We apply parameter-efficient LoRA adapters (Hu et al., 2022) to self-attention projections across layers 6–23 and optimize the joint objective in Eq. (3) at the Layer 8 hidden state.

3.4 SAE SHADOW ANCHORING

Directly anchoring dense hidden dimensions forces the network's residual stream into an artificial coordinate frame. We introduce *SAE shadow anchoring*: a sparse autoencoder is pretrained on Layer 8 activations and then strictly frozen. During adaptation, supervision is applied directly to the three SAE latent coordinates exhibiting highest mutual correlation with RGB channels on a held-out calibration split ($z_{2860}, z_{6432}, z_{5692}$). Gradients flow backward through the frozen SAE encoder and backbone into the LoRA adapters (Figure 1); neither the base model weights nor the SAE parameters are modified by the anchor loss.

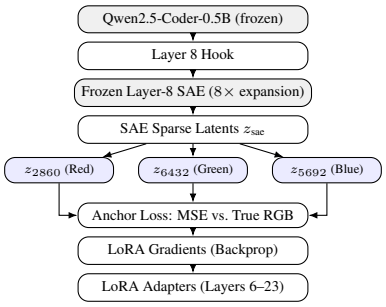

Figure 1: SAE shadow anchoring architecture. Both the backbone and the SAE are frozen. Gradients from the Layer 8 anchor loss propagate through the frozen SAE encoder into LoRA adapters in preceding layers (6–8), while the joint autoregressive LM loss backpropagates through all adapted layers (6–23).

4 Experimental Setup

Data. Experiments use a corpus of 100,000 raw CSS files (595M tokens) from public repositories. An AST validator verified that 92.4% pass full syntax validation including nested rules (`@media`, `@supports`). Data was formatted into non-overlapping windows of length 512, maintaining a 0.2 ratio of non-color to color-containing windows to preserve syntactic context. Branch 2 fine-tunes on a balanced subset of 5,000 windows (2.56M tokens) and evaluates on 500 held-out windows (256K tokens).

Architectures. Branch 1 trains a 6-layer, 384-dimensional, 6-head decoder transformer (38M parameters) with a 70,004-token vocabulary featuring 256 atomic hex digit pairs (`00`–`ff`). Branch 2 builds on Qwen2.5-Coder-0.5B (24 layers, 896 hidden dimensions, 152K vocabulary). The Layer 8 SAE expands 896 dense dimensions to 7,168 dictionary latents ($8\times$). Under moderate L_1 penalty ($\lambda = 8 \times 10^{-3}$), the autoencoder operates with dense dictionary activation ($L_0 \approx$ 2,600–3,500, $\sim$ 37–48% of latents), prioritizing broad gradient routing over extreme bottlenecking.

Training Hardware. Branch 2 LoRA fine-tuning uses AdamW (lr $= 2 \times 10^{-4}$, effective batch size 8, 3 epochs) with a depth-decaying rank schedule: $r = 16, \alpha = 32$ (layers 6–8), $r = 8, \alpha = 16$ (9–12), $r = 4, \alpha = 8$ (13–18), and $r = 2, \alpha = 4$ (19–23) across self-attention projections. To eliminate confounding factors, all Branch 2 phases shared seed 42, identical sample ordering, identical mini-batch sequences, and identical initial adapter weights. Compute: single consumer NVIDIA GPU (6GB VRAM, CUDA 12). Branch 1 pretraining took 10 CPU hours; Branch 2 fine-tuning took ~ 1.5 hours per phase.

Evaluation Sets. Held-out OOD evaluation uses 200 synthetic hex codes sampled uniformly across $[0, 1]^3$ absent from training data. Probing evaluation uses 5,013 token activations (80/20 train/test split).

5 Results

5.1 Branch 1: RGB Anchoring During Pretraining

Anchoring coordinates $\{0, 1, 2\}$ to RGB during pretraining yielded near-perfect training correlation ($r = 0.986$ for Red and Green, 0.985 for Blue). On the held-out set of 200 synthetic out-of-distribution hex codes, the model achieved pooled $r = 0.996$. This confirms that atomic digit-pair tokenization enables the network to learn a general compositional hex-to-value mapping rather than memorizing surface lookups.

Testing unconstrained HSL emergence against a permutation null baseline. We tested whether unanchored dimensions spontaneously organize into derived color coordinates (Hue, Saturation, Lightness) across 625 CSS `hsl()` calls. While peak correlations reached $|r| = 0.477$ (dim 330, exceeding a 500-shuffle null baseline $\mu = 0.108$, $p < 0.001$), an *unanchored baseline checkpoint* exhibited comparable baseline alignment ($|r| = 0.364$), and multivariate ridge regression yielded negative test R^2 ($\sin H : -1.55, S : -0.30, L : -2.61$). Spontaneous HSL emergence cannot be confirmed, serving as an informative negative control.

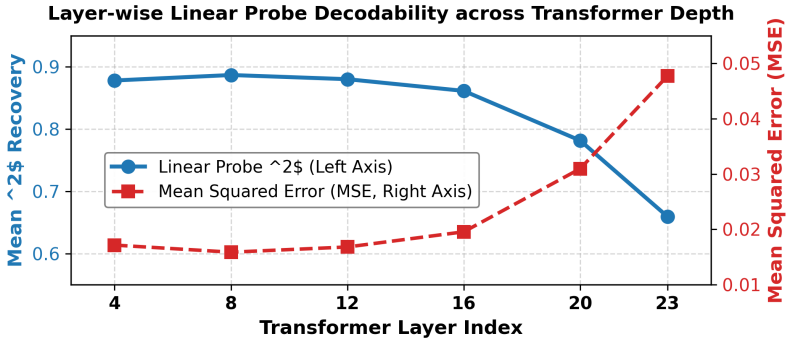

Figure 2: Layer-wise linear probe R^2 and mean squared error (MSE) for RGB recovery from unanchored Qwen2.5-Coder-0.5B hidden states. Recovery is highest on an intermediate plateau (Layers 4–12, peaking at Layer 8 of 24).

Table 1: Latent concentration in the frozen Layer 8 SAE evaluated on unadapted base model activations. Selected candidate latents explain substantial variance prior to adaptation ($R^2 = 0.2673$), while non-anchored latents preserve distributed representation capacity ($R^2 = 0.8271$).

Latent Subset Evaluated	**Latent Count**	**Probe R^2**	**Probe MSE**	**Avg Pearson r**
Full Layer 8 SAE Latents	7,168	**0.9473**	**0.0071**	**0.9733**
Non-anchored Latents	7,165	0.8271	0.0231	0.9095
Selected Target Latents (pre-anchoring: $z_{2860}, z_{6432}, z_{5692}$)	3	0.2673	0.0983	0.4828
Random 3-Latent Control ($z_{100}, z_{200}, z_{300}$)	3	0.0022	0.1338	0.0472

Extended anchoring (metadata and value types). Reserving dimensions 100–249 for CSS property names and 250–279 for value types (Phase 0.5) yielded compositional activations over shorthands. On 200 held-out files (440K tokens), this improved compression from 1.267 to 0.939 bits/byte (-25.9%) and increased next-token top-1 accuracy to 54.56% ($+2.28$ pp over standard BPE).

5.2 Branch 2: Anchoring a Pretrained Code Language Model

5.2.1 Layer Search

Sweeping a linear probe across six sampled layers of unanchored Qwen2.5-Coder-0.5B reveals an intermediate semantic plateau spanning Layers 4–12 ($R^2 \in [0.878, 0.894]$, peaking at Layer 8 with $R^2 \approx 0.89$; $R^2 = 0.8868$ on the layer sweep and 0.8944 on the selectivity split, MSE ≈ 0.016), before steadily decaying in deeper layers to $R^2 \approx 0.659$ at Layer 23 (Figure 2). This confirms the intermediate-layer hypothesis: middle layers develop rich semantic geometry before final layers specialize toward vocabulary logits (Tenney et al., 2019; Skean et al., 2025). Per-channel Layer 8 linear recovery reaches Pearson $r = 0.937$ (Red), 0.942 (Green), and 0.950 (Blue) (Appendix Figure 5).

5.2.2 Sparse Autoencoder Analysis

Linearly probing the frozen 7,168-dimensional Layer 8 SAE latents on unadapted base model activations achieves $R^2 = 0.9473$, exceeding raw hidden activations by $+0.0605$. Evaluating probe performance across latent subsets (Table 1) reveals that probing strictly the 3 candidate latents selected for subsequent anchoring ($z_{2860}, z_{6432}, z_{5692}$) yields $R^2 = 0.2673$ prior to adaptation. Probing the remaining 7,165 unanchored latents still recovers $R^2 = 0.8271$, confirming that while selected latents capture initial concept alignment, the model maintains a distributed representation across the dictionary, avoiding catastrophic bottlenecks.

5.2.3 Anchoring vs. Language Modeling Trade-off

Table 2 compares five configurations: the unadapted base model, final-layer anchoring (Phase 2), Layer 8 raw hidden anchoring (Phase 3), an unconstrained LoRA control fine-tuned without anchor loss (Phase 3-Verify), and Layer 8 SAE shadow anchoring (Phase 4). While a post-hoc linear

Table 2: Comparison across anchoring configurations (mean ± standard deviation across 5 independent random evaluation splits under training seed 42; deltas computed per split before rounding). Phases 3, 3-Verify, and 4 share identical LoRA capacity (layers 6–23) and training budgets. Phase 3-Verify provides an unconstrained domain-adapted control, cleanly isolating the interpretability tax from domain adaptation gains.

Metric / Setup	**Base Model** (Unmodified)	**Phase 2** (L23 Anchor)	**Phase 3** (L8 Anchor)	**Phase 3-Verify** (Unconstrained)	**Phase 4** **(L8 SAE Shadow)**
Representation Space	Native	Raw Hidden	Raw Hidden	Raw Hidden	SAE Latents
LoRA Target Layers	None	20–23	6–23	6–23	6–23
Anchor Depth	None	Layer 23	Layer 8	None	Layer 8 (via SAE)
LM Accuracy (%)	71.88 ± 0.75	71.55 ± 0.74 (-0.33)	73.47 ± 0.72 ($+1.59$)	73.74 ± 0.72 ($+1.86$)	$\mathbf{73.64 \pm 0.72}$ ($\mathbf{+1.76}$)
LM Loss (nats)	2.314 ± 0.119	1.326 ± 0.052	1.099 ± 0.046	1.088 ± 0.046	$\mathbf{1.092 \pm 0.046}$
Color MAE	0.8336 ± 0.0078	0.0750 ± 0.0032	0.0589 ± 0.0010	0.8369 ± 0.0080	$\mathbf{0.0305 \pm 0.0010}$
Pearson r (RGB)	-0.044 ± 0.017	0.9237 ± 0.0182	0.9689 ± 0.0023	-0.044 ± 0.017	$\mathbf{0.9871 \pm 0.0022}$

probe ($896 \to 3$) on the unadapted base model achieves $r \approx 0.94$ across Layer 8 activations, direct readout along arbitrary canonical basis coordinates yields chance correlation ($r = -0.044$). Concept anchoring retrofits a designated, fixed coordinate interface that requires no task-specific probe training ($r = 0.9871 \pm 0.0022$, MAE $= 0.0305 \pm 0.0010$).

Isolating the interpretability tax. Comparing the unconstrained control (Phase 3-Verify, $73.74 \pm 0.72\%$ LM accuracy) to the vanilla base model ($71.88 \pm 0.75\%$) shows that the $+1.86$ percentage-point gain reflects domain adaptation. Crucially, Phase 3-Verify isolates the true *interpretability tax*: directly constraining the dense residual stream at Layer 8 incurs a -0.26 ± 0.03 percentage-point penalty relative to unconstrained fine-tuning (73.47% vs. 73.74%, one-sample t-test $p = 6.7 \times 10^{-5}$ under evaluation noise). In contrast, SAE shadow anchoring cuts this tax by more than half to -0.10 ± 0.02 percentage points (73.64% vs. 73.74%, $p = 1.5 \times 10^{-4}$), yielding a statistically significant accuracy advantage over raw residual anchoring ($+0.17 \pm 0.04$ percentage points, paired t-test $t = 9.46$, $p = 0.0007$) while achieving superior continuous precision ($r = 0.9871 \pm 0.0022$, MAE $= 0.0305 \pm 0.0010 \approx 7.8/255$ intensity units).

We highlight two methodological considerations. First, while these 5 evaluation trials confirm that the $+0.17$ percentage-point advantage exceeds evaluation data-sampling noise ($p < 0.001$), verifying that this gain persists across independent training runs with varying fine-tuning seeds remains to be tested in future multi-seed training runs. Second, whereas Phase 3 anchors arbitrary canonical basis directions (e_0, e_1, e_2), shadow anchoring routes through aligned encoder dictionary directions. An important ablation is anchoring raw hidden states along linear-probe directions without an SAE to cleanly separate dictionary expansion from direction alignment. Furthermore, while the unadapted base model exhibits high cross-entropy loss (2.314 ± 0.119 nats) due to uncalibrated probability mass on tail tokens in domain CSS syntax, trailing LoRA adaptation rapidly calibrates the distribution (dropping loss to 1.326 ± 0.052 nats in Phase 2), even as forcing coordinates 0–2 at Layer 23 slightly constrains top-1 token precision ($71.55 \pm 0.74\%$ vs. $71.88 \pm 0.75\%$).

5.2.4 Causal Intervention, Generation Steering, and Specificity Control

To verify that anchored coordinates causally govern outputs, we implemented Layer 8 forward pre-hooks (Meng et al., 2022; Turner et al., 2024) on the Phase 4 adapted model. In color prompts, the hook fires at the token initiating a color literal (#), injecting steering vectors $\Delta h = \alpha \cdot z_{\text{target}} W_{\text{dec}}$ through the frozen SAE decoder, where z_{target} is a 3-vector (e.g., $+1.0$ for target and -0.5 for off-target channels).

Targeted color steering and dose-response. Across 10 CSS declaration prompts and calibrated scales ($\alpha \in \{5, 10, 15\}$), clamping the Green latent while suppressing Red and Blue shifts mean generated Green monotonically from 0.2561 (baseline) to 0.4659 at $\alpha = 10$ ($+81.9\%$) and 0.5588 at $\alpha = 15$ ($+118.2\%$ relative increase; Figure 3a). Across all 180 generation passes (6 steering conditions $\times$ 3 clamping levels $\alpha \in \{5, 10, 15\} \times 10$ test prompts), 100% of completions maintained valid CSS syntax (Appendix Table 6). However, intermediate residual intervention does not act as an isolated notch filter: non-target channels also rise with α (Red: $+0.121$ at $\alpha = 10$, $+0.243$ at $\alpha = 15$; Blue: $+0.179$ at $\alpha = 10$, $+0.361$ at $\alpha = 15$), reflecting broader color-subspace co-elevation and luminance shifts during downstream decoding. In Figure 3b, clamping individual latents distinctly alters output distributions, though single-coordinate clamping does not yield isolated monochrome primaries (e.g., Red clamping drives darkened, low-intensity palettes with $R = 0.021$ rather than

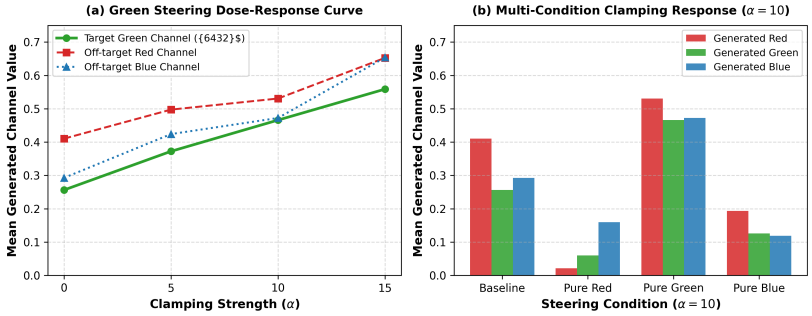

Figure 3: Causal steering via Layer 8 SAE intervention on Phase 4 adapted Qwen2.5-Coder-0.5B. (a) Dose-response curve showing continuous elevation of target Green from 0.2561 (baseline) to 0.4659 at $\alpha = 10$ (+81.9%) and 0.5588 at $\alpha = 15$ (+118.2%), alongside broad subspace co-elevation of non-target channels. (b) Multi-condition clamping response at $\alpha = 10$ across primary color conditions (Baseline, Pure Red, Pure Green, Pure Blue).

Table 3: Comparison of causal steering mechanisms at Layer 8. SAE shadow clamping monotonically elevates target channels with 100% syntax validity, accompanied by subspace co-activation. Dense DiffMean / CAA suffers from high collinearity ($\cos = 0.608$), inducing off-target distortion and syntax degradation (80% at $\alpha = 15$).

Steering Mechanism	Target Channel Output	Off-Target Channel Shifts	Syntax Validity	Code Intrusion
Baseline (Unsteered)	Green: 0.256 / Red: 0.410 / Blue: 0.293	—	100%	0/10
Dense DiffMean / CAA ($\alpha = 10$)	Green: 0.067 (−73.8%)	Red: −0.110 / Blue: −0.093	100%	1/10
Dense DiffMean / CAA ($\alpha = 15$)	Green: 0.058 (−77.2%)	Red: −0.035 / Blue: +0.082	80.0%	2/10
SAE Shadow Clamping ($\alpha = 10$)	**Green: 0.466 (+81.9%)**	Red: +0.121 / Blue: +0.179	**100%**	**0/10**
SAE Shadow Clamping ($\alpha = 15$)	**Green: 0.559 (+118.2%)**	Red: +0.243 / Blue: +0.361	**100%**	**0/10**

pure `#ff0000`), illustrating the nonlinear readout dynamics of trailing transformer layers (Layers 9–23).

Specificity on general code. Evaluating the steered model ($\alpha = 10$) on non-color algorithmic prompts spanning Python (quicksort, binary search, matrix multiplication), JavaScript (tax calculation, discounts), and CSS layout (flex, grid) confirmed that steering introduces zero spurious color literals into general code (0/10 intrusion). On these prompts, the hook intervened unconditionally at the final prompt token. While 100% algorithmic syntax was preserved, boundary residual perturbation induced minor lexical variations in generated completions (e.g., quicksort branch threshold shifting from ≤ 1 to ≤ 2, and flex alignment adjustments; Appendix Table 6).

Comparison with training-free contrastive steering (DiffMean / CAA). We benchmarked against Contrastive Activation Addition (Zou et al., 2023; Rimsky et al., 2024). Difference-of-means vectors $\hat{v}_c$ computed on Layer 8 activations exhibit strong mutual collinearity ($\cos(\hat{v}_R, \hat{v}_G) = 0.6082$), conflating color intensity with shared syntactic features. As shown in Table 3, DiffMean steering fails: steering along $\hat{v}_R$ at $\alpha = 10$ suppresses Red ($0.410 \rightarrow 0.194$) while elevating off-target Blue (0.376), and Green steering suppresses target Green to 0.067 (−73.8%). At $\alpha = 15$, DiffMean degrades syntax, dropping valid hex rate to 80.0%. In contrast, SAE shadow clamping achieves monotonic target elevation (+118.2% at $\alpha = 15$) with 100% syntax validity, though accompanied by off-target channel co-elevation.

5.2.5 Syntax-Variant Generalization Across Languages

To verify that shadow anchoring learns continuous semantics rather than overfitting to standard CSS declaration templates (`color: #...`), we tested Layer 8 representations of the Phase 4 anchored model on 100 uniformly sampled synthetic colors across 6 distinct syntactic paradigms (Table 4). Uniform sampling across the full color cube is harder than the natural corpus distribution in Table 2, explaining $r = 0.781$ on standard CSS versus $r = 0.987$ on in-distribution data. While CSS custom properties, runtime `var()` fallbacks, and HTML inline styles represent in-domain syntactic variants present in diverse CSS corpora, JavaScript object literals and Python dictionaries represent zero-shot out-of-domain contexts without any fine-tuning on those languages. Across all 6 paradigms, the primary color latent maintained robust linear correlation ($r \in [0.710, 0.820]$, probe $r \geq 0.516$),

Table 4: Syntax-variant generalization of Layer 8 representations of the Phase 4 anchored model across 6 paradigms (100 uniformly sampled synthetic RGB colors, harder than the natural color distribution in Table 2; Red channel shown). Representations transfer across in-domain CSS variants and zero-shot across JavaScript and Python literals ($r \in [0.71, 0.82]$).

Syntactic Paradigm	Representative Template	Probe r_{Red}	SAE Latent r_{Red}	Syntax Valid
CSS Standard Property	`div { color: #...; }`	0.724	0.781	100%
CSS Custom Property	`:root { --theme-accent: #...; }`	0.805	0.763	100%
CSS `var()` Fallback	`button { bg: var(--c, #...); }`	0.516	0.710	100%
HTML Inline Style	`<span style="color: #...">`	0.753	0.820	100%
JavaScript Object	`const t = { primaryColor: "#..." };`	0.742	0.810	100%
Python Dictionary	`config = {"brand_color": "#..."}`	0.789	0.799	100%

Table 5: Hewitt & Liang probe selectivity on unadapted base model Layer 8 activations and frozen SAE dictionary across representation spaces (prior to Phase 4 anchoring). Both raw hidden states and SAE latents collapse on shuffled control targets, establishing large positive selectivity margins ($\Delta R^2 > +0.27$ to $+1.16$). These results are consistent with continuous geometric encoding in the pre-trained base model prior to shadow anchoring.

Representation Substrate (Base Model)	True Task R^2	Shuffled Task R^2	Random Task R^2	Selectivity (ΔR^2)
Raw Layer 8 Hidden (896 dims)	**0.8944**	-0.1760	-0.2150	**+1.0704**
Selected 3 Pre-Anchoring SAE Latents (z_{RGB})	**0.2673**	-0.0065	-0.0012	**+0.2737**
Full SAE Dictionary (7,168 dims)	**0.9344**	-0.2266	-0.3081	**+1.1610**

confirming that the anchored representation binds the underlying continuous numeric ground truth across disparate programming contexts rather than memorizing surface syntax templates.

5.2.6 Evaluating Representation Shortcuts: Hewitt & Liang Probe Selectivity

To evaluate whether the base model's Layer 8 representations capture genuine continuous geometry rather than memorizing a discrete token lookup table prior to anchoring, we implemented Hewitt & Liang control tasks (Hewitt & Liang, 2019). Linear probes were trained on unadapted base Layer 8 activations and its frozen pre-trained SAE to predict: (i) true RGB targets, (ii) randomly permuted/shuffled RGB targets per instance, and (iii) uniform random targets (Table 5 and Figure 4a).

On raw Layer 8 activations of the unadapted base model, probe performance collapses from $R^2 = 0.8944$ on the true task (consistent with the $R^2 \approx 0.89$ probe baseline in Section 5.2.1) to -0.1760 under shuffled controls ($\Delta R^2 = +\mathbf{1.0704}$). For the 3 selected pre-anchoring SAE latents, the linear probe collapses from $R^2 = 0.2673$ (reflecting pre-anchoring correlation $r \approx 0.48$) to -0.0065 ($\Delta R^2 = +\mathbf{0.2737}$). Across the full 7,168-dimensional frozen SAE dictionary, performance collapses from $R^2 = 0.9344$ (closely matching the $R^2 = 0.9473$ full-dictionary probe in Table 1) to -0.2266 ($\Delta R^2 = +\mathbf{1.1610}$). This decisive collapse on control tasks is consistent with continuous geometry and rules out fitting of arbitrary targets in the base model's pre-existing representations; a type-level control would more directly evaluate discrete lookup memorization.

5.2.7 The Format Gap: Multi-Token Binding Latency

Evaluating recovery across token formats reveals that probes recover consolidated literals ($R^2 = \mathbf{0.9421}$, $N = 807$) substantially better than multi-token syntax (`rgb()`, `hsl()`; $R^2 = \mathbf{0.6640}$, $N = 128$) at Layer 8 (Figure 4b). This $+0.2781$ gap highlights multi-token binding latency: consolidated literals are more linearly decodable at Layer 8, whereas functional formats require deeper self-attention to compose distributed sub-tokens into unified concepts.

6 Discussion

The central takeaway of this work is that concept anchoring can survive inside the multi-layer attention and residual dynamics of an already-pretrained transformer with a small measured cost to language modeling. Whereas Fraser & Wielopolski (2025) studied small autoencoders without attention, our results demonstrate that LoRA-mediated auxiliary supervision retrofits continuous grounded concepts directly into LLM residual streams.

Second, SAE shadow anchoring introduces a distinct architectural paradigm relative to recent concept-aligned sparse autoencoders (Yang et al., 2026; Kulkarni et al., 2026): rather than adapting

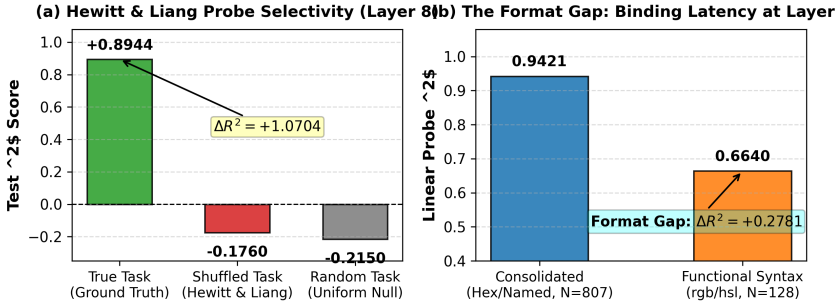

Figure 4: Representation geometry diagnostics at Layer 8. (a) Hewitt & Liang probe selectivity: probes collapse from $R^2 = +0.8944$ on true targets to -0.1760 on shuffled controls ($\Delta R^2 = +1.0704$). (b) The format gap: consolidated literals are more linearly decodable at Layer 8 ($R^2 = 0.9421$) than multi-token functional syntax ($R^2 = 0.6640$), reflecting multi-token binding latency.

the SAE dictionary, we *freeze the SAE as an audited dictionary and backpropagate supervision into the base model's adapters*. We hypothesize this reduces feature interference, conferring a $+0.17$ percentage point advantage in LM accuracy tax over raw hidden anchoring ($p < 0.001$ under a fixed training seed; separating dictionary expansion from direction alignment is left to future work, see §5.2.3). Crucially, Layer 8 anchoring exploits the intermediate semantic peak (Tenney et al., 2019; Skean et al., 2025) for causal intervention.

7 Limitations

Model scale and domain scope. Evaluations focused on Qwen2.5-Coder-0.5B and continuous RGB color; scaling to larger models (1.5B–7B, LLaMA, DeepSeek) and discrete bottlenecks (AST depth, types) warrants future study. **SAE density and benchmarks.** The SAE operates with moderate density ($L_0 \approx 2{,}600$–$3{,}500$ of 7,168 latents); TopK/JumpReLU architectures could enforce stricter sparsity. Multi-seed training, anchoring raw hidden states along linear-probe directions without an SAE, minimal supervision ($< 0.1\%$), and multi-language benchmarks (HumanEval, MBPP) remain open frontiers.

8 Conclusion

We investigated retrofitting grounded concepts into latent coordinates of pretrained transformers using continuous RGB color in code as a machine-checkable testbed. We showed that: (1) structural digit-pair tokenization enables compositional OOD generalization ($r = 0.996$); (2) intermediate Layer 8 forms a natural semantic peak ($R^2 \approx 0.89$); (3) SAE shadow anchoring achieves near-perfect continuous recovery ($r = 0.9871$, MAE$= 0.0305$) with a small LM cost ($+0.17$ pp advantage over raw anchoring, $p < 0.001$ under a fixed training seed); (4) latent clamping causally steers generated code monotonically ($+118.2\%$ at $\alpha = 15$) with 100% syntax validity, though with off-target co-elevation; and (5) syntactic transfer and Hewitt–Liang controls are consistent with continuous geometric encoding. These results demonstrate that training-time concept structuring can be retrofitted with small measured cost in this setting.

Reproducibility Statement

To ensure the reproducibility of all empirical results, comprehensive specifications of our architectures, training procedures, and evaluation protocols are provided throughout the paper. Section 4 and Appendix A provide complete architectural hyperparameters, LoRA rank-decay schedules, SAE training configurations, and anchored latent indices. Section 5.2.3 details the evaluation benchmark across 5 independent evaluation splits under a fixed training seed along with exact statistical significance tests. Furthermore, complete, self-contained, and anonymized source code covering the Branch 1 tokenization pipeline, Branch 2 LoRA fine-tuning, SAE shadow backpropagation, causal steering pre-hooks, and benchmark evaluation scripts is provided in the supplementary material package.

Ethics Statement

This work adheres to the ICLR Code of Ethics. Our experiments utilize publicly available, permissively licensed open-source CSS repositories from GitHub and do not involve human subjects, personally identifiable information (PII), or private user data. The primary objective of concept anchoring and SAE shadow anchoring is enhancing the transparency, interpretability, and steerability of language models, mitigating opaque internal representations without introducing harmful capabilities or deceptive behaviors.

AI Use Statement

We used Gemini to help write and debug experiment code, Gemini to polish the writing of the experiments and other sections, and Gemini to draft portions of the text. The authors designed the study, ran and verified all experiments, checked every reference, and take full responsibility for the content.

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color. In Arianna Bisazza and Omri Abend (eds.), *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 109–132, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.9. URL https://aclanthology.org/2021.conll-1.9/.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644, 2016. URL https://api.semanticscholar.org/CorpusID:9794990.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=awIpKpwTwF.

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

Zhi Chen, Yijie Bei, and Cynthia D. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2:772 – 782, 2020. URL https://api.semanticscholar.org/CorpusID:211031886.

Hoagy Cunningham, Aidan Ewart, Logan Riggs Smith, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *ArXiv*, abs/2309.08600, 2023. URL https://api.semanticscholar.org/CorpusID:261934663.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Thomas Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Baker Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Chris Olah. Toy models of superposition. *ArXiv*, abs/2209.10652, 2022. URL https://api.semanticscholar.org/CorpusID:252439050.

Sandy Fraser and Patryk Wielopolski. Sparse concept anchoring for interpretable and controllable neural representations. *ArXiv*, abs/2512.12469, 2025. URL https://api.semanticscholar.org/CorpusID:283895883.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *ArXiv*, abs/1909.03368, 2019. URL https://api.semanticscholar.org/CorpusID:202538609.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Jun Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:51737170.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Akshay Kulkarni, Tsui-Wei Weng, Vivek Narayanaswamy, Shusen Liu, Wesam A. Sakla, and Kowshik Thopalli. Interpretable and steerable concept bottleneck sparse autoencoders, 2026. URL https://arxiv.org/abs/2512.10805.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Vi'egas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ArXiv*, abs/2210.13382, 2022. URL https://api.semanticscholar.org/CorpusID:253098566.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-h6WAS6eE4.

Fabian Morelli, Arnas Uselis, Ankit Sonthalia, and Seong Joon Oh. Sparse autoencoders enable robust and interpretable fine-tuning of clip models, 2026. URL https://arxiv.org/abs/2605.15961.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=9XFSbDPmdW.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2/.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

Alethea Power, Yuri Burda, Harrison Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *ArXiv*, abs/2201.02177, 2022. URL https://api.semanticscholar.org/CorpusID:245769834.

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

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

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, 2026. URL https://arxiv.org/abs/2605.29358.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

{Alexander Matt} Turner, Lisa Thiergart, Gavin Leech, David Udell, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. Workingpaper, arXiv.org, United Kingdom, June 2024.

Minglai Yang, Xinyu Guo, Zhengliang Shi, Jinhe Bi, Steven Bethard, Mihai Surdeanu, and Liangming Pan. Alignsae: Concept-aligned sparse autoencoders. *Transactions on Machine Learning Research*, 2026-August, 2026. ISSN 2835-8856. Publisher Copyright: © 2026, Transactions on Machine Learning Research. All rights reserved.

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 Troy Mallen, Steven Basart, Oluwasanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency. *ArXiv*, abs/2310.01405, 2023. URL https://api.semanticscholar.org/CorpusID:263605618.

A Additional Implementation Details and Qualitative Results

A.1 Per-Channel Probing Scatter at Layer 8

Figure 5 shows linear probe predictions versus ground-truth values across individual color channels at Layer 8 of unanchored Qwen2.5-Coder-0.5B, showing uniform linear decodability across Red, Green, and Blue.

A.2 Qualitative Examples of Causal Steering

Table 6 provides qualitative completions under Layer 8 causal steering ($\alpha = 10.0$) across color-sensitive CSS contexts and non-color algorithmic benchmarks, demonstrating target channel elevation and zero code intrusion.

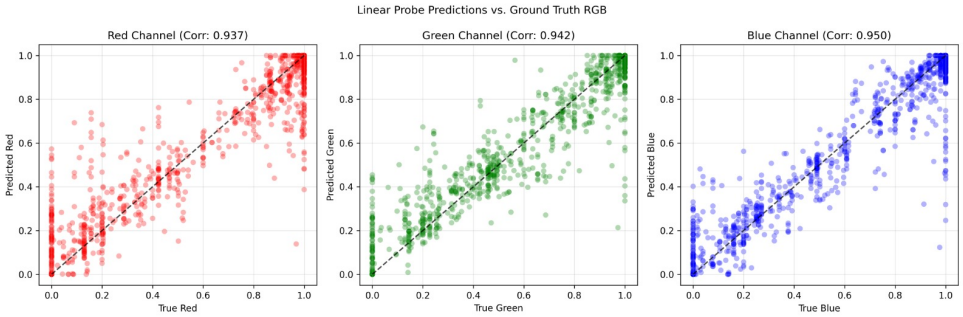

Figure 5: Linear probe predictions versus ground-truth RGB at Layer 8 of unanchored Qwen2.5-Coder-0.5B across individual color channels (Pearson $r = 0.937$ for Red, 0.942 for Green, 0.950 for Blue).

Table 6: Qualitative examples of causal steering at Layer 8 ($\alpha = 10.0$). Clamping color latents shifts generated color literals while preserving valid syntax, and produces zero spurious intrusions in general algorithmic code.

Prompt	**Baseline Completion**	**Steered Completion** ($\alpha = 10.0$)
Color-sensitive CSS contexts:		
`button.btn-primary { background-color: #`	`2196F3; color: #fff;` *(Unsteered: dominant blue)*	`00ff88; color: #000;` ***(Steered green: green-shifted)***
`.alert-box { border-left: 4px solid #`	`f00; border-right:...` *(Unsteered: dominant red)*	`f0f0f0; background:...` ***(Steered green: desaturated/lightened)***
Non-color algorithmic contexts (Specificity control):		
`def quicksort(arr):`	`if len(arr) <= 1: return arr` *(Standard recursion base)*	`if len(arr) <= 2: return arr` ***(Valid Python; 0% color)***
`const calculateTax = (amount, rate) =>`	`amount * (1 + rate);` *(Arithmetic computation)*	`amount * (1 + rate);` ***(Identical; 0% color)***
`div.container { display: flex;`	`flex-direction: column;` *(Standard CSS layout)*	`align-items: center;` ***(Valid CSS layout; 0% color)***

A.3 Hyperparameter and Architectural Details

- **Branch 1 Tokenizer & Architecture:** CSS AST tokenizer with 70,004 vocabulary tokens. 6 layers, 384 hidden dimensions, 6 attention heads (38M parameters). Pretrained using AdamW (lr $= 5 \times 10^{-4}$), context length 512, batch size 4 for 5,000 steps.
- **Branch 2 LoRA Configuration:** Rank-decay schedule: layers 6–8 ($r = 16, \alpha = 32$), layers 9–12 ($r = 8, \alpha = 16$), layers 13–18 ($r = 4, \alpha = 8$), layers 19–23 ($r = 2, \alpha = 4$), targeting `q_proj`, `k_proj`, `v_proj`, `o_proj` with dropout 0.05 (642,048 trainable parameters, ~ 0.64M, 0.13% of base model).
- **SAE Architecture & Training:** ReLU activation, tied unit-norm decoder; input 896, latents 7,168 (8×). Trained for 100,000 steps with Adam (lr $= 3 \times 10^{-4}$), L_1 penalty $\lambda = 8 \times 10^{-3}$, batch size 64. Anchored latent indices: 2860 (Red), 6432 (Green), 5692 (Blue).
- **Specificity Benchmark Prompts:** Python algorithms (`def quicksort(arr):`, `def binary_search(arr, target):`, `import numpy as np; def matrix_multiply(a, b):`), JavaScript routines (`const calculateTax = (amount, rate) =>`, `function getDiscount(price, rate)`), and CSS layout rules (`div.container { display: flex; }`, `table.data-grid { border-collapse: collapse; }`, `p.article-body { font-size:`

```
16px; }, ul.nav-menu { list-style:  none; }, section.hero-box {
box-sizing:  border-box; }).
```