# WHERE FEATURES ARE WRITTEN AND READ: LOCALIZING NATIVE TRANSFORMER CHANNELS FOR RESIDUAL DIRECTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Sparse autoencoders, probes, and steering methods find directions in a language model's activations that track concepts, and editing along them can steer the model's behavior. A direction does not say which of the model's own components build the feature or which carry it to behavior, so it is unclear whether an edit along the direction reflects how the model itself computes. We introduce feature-direction localization, which maps a supplied direction to the model's own multilayer perceptron (MLP) channels on both sides of it: *writers*, the channels that construct the feature, and *consumers*, the channels through which it reaches behavior. Across interpretable features in several language models, as little as 0.1% of the candidate MLP channels construct 90% of what the MLP layers contribute to a feature, and 0.06% of downstream channels account for 91% of the behavioral effect of steering along it. Connecting the two maps shows that the writers' behavioral effect depends on the same consumers that carry steering: holding those consumers at their clean activations removes much of it. An edit along the direction matched to the writers' change in the feature tests whether steering reproduces their behavioral effect: in one model the matched edit reproduces little of it, while the part of the writers' change that leaves the feature's coordinate unchanged reproduces 85%.

## 1 INTRODUCTION

Mechanistic interpretability describes model computation in two ways. Representation-level methods, such as sparse autoencoders (SAEs) and contrastive steering vectors, find directions in a model's activations that track semantic and grammatical concepts and can steer behavior (Huben et al., 2024; Bricken et al., 2023; Lieberum et al., 2024; Rimsky et al., 2024). Component-level methods trace the attention heads and neurons that implement a task (Conmy et al., 2023; Wang et al., 2022; Hanna et al., 2023). The two descriptions rarely meet. A feature direction says what is represented but not which native components build or use it, and circuit analyses built on learned features use those features as nodes in place of the model's own channels. Steering along a direction shows that the model responds to an edit, but it does not test whether the edit does what the model's own components do. The closest work localizes tasks in the native neuron basis (Arora et al., 2026) or builds new features from groups of coactivating neurons (Shafran et al., 2026); neither starts from a supplied direction and asks which native channels build it and which carry it to behavior.

We connect the two with *feature-direction localization*. Given a supplied residual direction, we find sparse sets of native multilayer perceptron (MLP) channels on both sides of it (Figure 1):

- *Writers* are upstream channels that construct the feature. Replacing their activations with means estimated on weak-feature inputs should remove the feature while changing little else.
- *Consumers* are downstream channels through which the feature reaches behavior. After an edit along the feature direction, holding their activations at their clean values should cancel the edit's behavioral effect.

MLP channels are a natural unit because feed-forward layers read and write residual directions (Geva et al., 2021; Arora et al., 2026). The interventions act on the original model, so the localized sets describe the model as trained rather than a surrogate. Connecting the two maps shows that native writers act through consumers discovered from steering, and that those consumers preferentially



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.

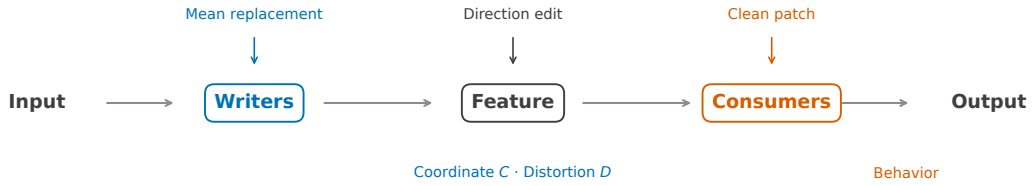



Figure 1: Writer and consumer localization. Writers are found by mean replacement and scored by coordinate construction $C$ and orthogonal distortion $D$; consumers are found by restoring clean activations after a direction edit and scored by how much of the edit's behavioral effect the restoration removes.

restore their own feature's edit. Steering and the native writers can still have sharply different behavioral effects, even though they act through the same consumers.

**Contributions.**

1. **Mapping supplied directions to native channels.** A protocol that locates a direction's upstream writers and downstream consumers, with every set evaluated by exact intervention on the original model.
2. **Sparse, behaviorally connected maps.** For syntactic, factual, numerical, and contrastive features in several language models, small channel sets construct much of the feature's MLP contribution and carry much of a steering edit's effect, and consumers found from steering also carry the native writers' effect, preferentially for their own feature.
3. **Testing steering against native computation.** Matching a direction edit to the writers' coordinate change shows when steering reproduces the writers' behavioral effect and when the coordinate-preserving part of their change reproduces it instead.

## 2 Method

### 2.1 Setup

Fix a residual-stream state $\mathbf{r}_L(x)$ at layer $L$ and token position $t$, and a supplied unit direction $\mathbf{d}$ along which to edit it. To localize writers, we specify a scalar coordinate that measures what they construct. For a direction without a separate readout, the natural choice is its projection $z(x) = \mathbf{d}^\top \mathbf{r}_L(x)$. More generally, we allow a supplied linear readout $z(x) = \mathbf{e}^\top \mathbf{r}_L(x) + b$; writer localization is relative to this chosen coordinate. For the contrastive direction studied here, reading and editing share an axis, whereas an SAE supplies an encoder readout and a decoder edit direction, as detailed below.

Each direction is paired with a task, and the *task endpoint* $B(x)$ is a scalar summary of the model's prediction on input $x$, such as the logit of the correct answer minus that of an incorrect one. MLP channel $(\ell, j)$ has activation $a_{\ell j}$ (written $a_j$ when the layer is clear) and adds $a_{\ell j}$ times its down-projection column to the residual stream. Channel interventions act on these native activations; direct edits act on the residual stream. Attention, normalization, and unselected channels remain active. Each case uses three disjoint input splits: discovery inputs for ranking channels and estimating references, a tuning split for selecting sets and budgets, and held-out inputs for evaluation.

**SAE instantiation: encoder readout, decoder edit.** For SAE features, $\mathbf{e}$ is the encoder row, $b$ its effective bias (including any input preprocessing, such as a subtracted decoder bias), and $\mathbf{d}$ the normalized decoder vector. We target the encoder coordinate $z$, because the encoder defines the SAE's activation of the feature: the feature fires when $z$ exceeds the SAE's threshold, so writers of $z$ are the channels that make the SAE register the feature. The decoder projection $p = \mathbf{d}^\top \mathbf{r}_L$ is an alternative target; it measures residual content along the feature's reconstruction direction, which also includes the projections of other active features and of the reconstruction error. Appendix E compares the two targets.

### 2.2 WRITER LOCALIZATION

Writer localization seeks channels whose replacement removes the supplied coordinate while changing little else in the residual stream. Let $\mathbf{r}_S$ be the feature-layer residual after replacing the activations of channel set $S$ with their reference means $\mu_j$, estimated on low-coordinate discovery inputs: feature-off inputs when available, otherwise the lowest-coordinate half. This replaces each example-specific activation with its low-coordinate reference rather than with zero. Let $\mathbf{r}_{\text{all}}$ be the residual after replacing all candidate channels. Construction measures the fraction of the all-candidate reference drop that $S$ removes, and distortion measures the collateral residual change outside the edit direction:

$$C(S) = \frac{\mathbb{E}_x\left[z_{\text{clean}}(x) - z_S(x)\right]}{\mathbb{E}_x\left[z_{\text{clean}}(x) - z_{\text{all}}(x)\right]}, \qquad D(S) = \mathbb{E}_x\left[\frac{\|P_{\mathbf{d}^\perp}(\mathbf{r}_{\text{clean}}(x) - \mathbf{r}_S(x))\|_2^2}{\|\mathbf{r}_{\text{clean}}(x)\|_2^2}\right], \tag{1}$$

where $z_S$ and $z_{\text{all}}$ are the coordinates of these residuals and $P_{\mathbf{d}^\perp}$ removes the component along $\mathbf{d}$. Normalizing by the all-candidate drop gives $C = 1$ the same reference meaning on every checkpoint, and a subset can exceed it through opposite-signed contributions. $D$ measures the residual change orthogonal to $\mathbf{d}$, which removing a feature should leave intact. It measures collateral change directly in the residual stream, a geometric proxy for effects on other features.

**The construction–distortion trade-off.** Channel effects can reinforce or cancel, so we evaluate whole sets. On the tuning split, we compare candidate sets by their construction error $|C(S) - 1|$ and distortion $D(S)$. The Pareto frontier contains sets no other candidate matches on both objectives and beats on one.

**Writer methods.** Each method produces one candidate set per budget $K$. We first consider two attribution-based rankings from prior work. The *coordinate ranking* applies attribution patching (Syed et al., 2024; Kramár et al., 2024) to the feature coordinate. It computes the per-example first-order attribution

$$c_j(x) = (a_j(x) - \mu_j)\,\frac{\partial z(x)}{\partial a_j}, \tag{2}$$

where the gradient passes through every layer between the channel and the feature layer, and keeps the $K$ channels with the largest mean over discovery inputs. *Endpoint attribution*, the same estimator applied to the task endpoint $B$, is a baseline asking whether task-relevant channels are writers.

We incorporate distortion in two ways. The *per-channel penalty score* ranks channels by first-order construction minus $\lambda$ times their individual orthogonal distortion. Construction uses the reading coordinate $z$ or the projection $p$ onto the edit direction, which are the encoder and decoder targets for SAE features; the penalty grid includes $\lambda = 0$. This scores channels independently and cannot account for cancellation within a set. *Exact refinement* starts from the top-$K$ coordinate set and screens swaps using first-order estimates. A swap is accepted only when an exact tuning-split intervention does not lower $C$, does not raise $D$, and improves at least one. The resulting set is no worse in $C$ and $D$ than the starting set on tuning data, but increasing $C$ above 1 can worsen $|C - 1|$. Appendix A gives the penalty grid and search settings.

**Writer set selection.** We select one writer set per checkpoint, the operating point, at the knee of the tuning-split construction–distortion frontier, using the coordinate, per-channel penalty, and refined candidates across budgets: after scaling both objectives to $[0, 1]$ over the frontier, the knee is the frontier point farthest from the line joining its two ends. The knee balances the two objectives without a construction threshold, and selection uses no behavioral objective. Appendix A.3 examines the knee's sensitivity to the candidate methods and to the selection criterion.

### 2.3 CONSUMER LOCALIZATION

Consumer localization identifies channels whose restoration reverses a direction edit's behavioral effect. For an edit $\mathbf{r}_L \mapsto \mathbf{r}_L + \alpha\mathbf{d}$ (amplitudes and hooks in Appendix G), let $h_j^0$ and $h_j^1$ be the clean and edited activations of downstream channel $j$, and $B^0$ ($= B_{\text{clean}}$) and $B^1$ the corresponding task endpoints. We restore a candidate set to its clean activations during the edited run, giving endpoint $B^2$, and measure signed attenuation and absolute restoration:

$$A = 1 - \frac{\mathbb{E}[B^2 - B^0]}{\mathbb{E}[B^1 - B^0]}, \qquad A_{\text{abs}} = 1 - \frac{\mathbb{E}|B^2 - B^0|}{\mathbb{E}|B^1 - B^0|}. \tag{3}$$

$A = 1$ means restoration cancels the mean signed effect. $A$ can exceed 1 or be negative, and per-example errors of opposite sign can cancel in it; $A_{\text{abs}}$ uses absolute errors, which cannot.

**Consumer rankings.** We compare five consumer rankings. The main one is attribution patching (Syed et al., 2024), with the edited run in place of a corrupted input: over the $N$ discovery inputs,

$$s_j^{\text{cons}} = \frac{1}{N} \sum_x o_x \left(h_j^1 - h_j^0\right) \frac{\partial B^1}{\partial h_j^1}, \tag{4}$$

summed over token positions, with $o_x = \text{sign}(B^1(x) - B^0(x))$ (Qwen uses the sign-specific protocol in Appendix G.1). The others are eight-step integrated gradients (IG), as in EAP-IG (Hanna et al., 2024); response magnitude $|h_j^1 - h_j^0|$; and gradient times activation, $o_x h_j^1 \, \partial B^1 / \partial h_j^1$. Screened exact patching measures each channel's effect by a full forward pass within a pool nominated by the other four rankings and keeps the $K$ largest (Appendix G).

**Consumer budget selection.** Using the attribution ranking, we measure on the tuning split the remaining effect $s\,\mathbb{E}[B^2 - B^0]$, signed by the direction $s$ of the edit's mean effect, at each budget. Budgets from the first at which it changes sign overshoot the clean endpoint, so we take the smallest earlier $K$ within one standard error of the smallest remaining effect among them. Every ranking is then compared at this budget. Qwen, whose signed effects cancel across its two probe wordings, instead takes the budget with the largest absolute restoration for each steering sign (Appendix G).

### 2.4 Composition and behavioral handles

**Do writers act through consumers?** We test whether consumers discovered from direction edits also transmit native writer effects, by ablating the writer set $S$ while holding the consumer set $\mathcal{C}$ at its clean activations. Let $B_S$ be the endpoint after writer ablation alone and $B_{S,\mathcal{C}}$ the endpoint with consumer restoration as well. The fraction of the mean writer effect removed by restoration is

$$M_W = 1 - \frac{\mathbb{E}[B_{\text{clean}} - B_{S,\mathcal{C}}]}{\mathbb{E}[B_{\text{clean}} - B_S]}. \tag{5}$$

$M_W = 1$ means restoration cancels the mean signed writer effect; equal-size random sets are the control.

**Is the supplied direction a faithful behavioral handle?** We compare writer ablation with a direct edit that produces the same coordinate change but acts only along $\mathbf{d}$. For $\mathbf{e}^\top \mathbf{d} \neq 0$, this edit is

$$\mathbf{r}_{\text{direct}} = \mathbf{r}_{\text{clean}} + \frac{z_S - z_{\text{clean}}}{\mathbf{e}^\top \mathbf{d}} \, \mathbf{d}.$$

An intervention's effect is the clean endpoint minus the intervened endpoint, so the writer and direct effects are $\Delta B_W = \mathbb{E}[B_{\text{clean}} - B_S]$ and $\Delta B_{\text{dir}} = \mathbb{E}[B_{\text{clean}} - B_{\text{direct}}]$. We compare the mean effects and calibrate the direct edit's mean absolute effect against random-direction edits of the same per-example norm. We call the direction a *faithful handle* where its edit clearly exceeds these controls and approximates the writer effect. Per example, we compare signs and the relative discrepancy $\sum_x |\Delta B_W(x) - \Delta B_{\text{dir}}(x)| / \sum_x |\Delta B_W(x)|$, which is 0 for identical effects and 1 for a null edit (Appendix F.2).

## 3 Experimental Setup

Table 1 lists the cases, which span five models from 70M to 8B parameters (Radford et al., 2019; Biderman et al., 2023; Team et al., 2024; 2025; Yang et al., 2025) and test localization on published features, an established task circuit, a newly characterized feature, and a non-SAE steering direction. On GPT-2 the consumers can be compared with the published Greater-Than neurons. On Gemma-3 we select the feature and define the task ourselves, where no circuit is known, and the base and IT checkpoints share feature, task, layer, and candidate pool.

Table 1: Cases. Each supplied direction is paired with a task whose endpoint the interventions move; Appendix M gives hooks, prompts, and endpoints.

| Checkpoints | Direction | Type | Task |
|---|---|---|---|
| GPT-2 Small | F2448 | SAE | Greater-Than year comparison |
| Pythia-70M | F14719, plural subject | SAE | Subject–verb agreement |
| Gemma-2-2B base*, IT | F7957, known entity | SAE | Claimed knowledge of an entity |
| Gemma-3-1B base, IT | F4646, demonstrative number | SAE | Demonstrative–verb agreement |
| Qwen3-8B | Value Axis | Contrastive | Confidence probes; math and code |

*Writer results only: it responds little to steering on the chat endpoint.

Table 2: Writer operating points (knee of the tuning-split frontier) on held-out data, next to the mean of five equal-size random sets. $\Delta B_W$ is in each checkpoint's endpoint units (Qwen: confidence probes). †Pooled over three datasets. Refined sets with no accepted swap are labelled coordinate. Penalty sets come from the per-channel penalty score, with construction measured on $z$ or $p$ and penalty strength $\lambda$.

| | | | Construction $C$ | | $10^3\ D$ | | $\Delta B_W$ | |
|---|---|---|---|---|---|---|---|---|
| Checkpoint | Set | $K$ | Writers | Random | Writers | Random | Writers | Random |
| GPT-2 Small | coordinate | 3 | 0.61 | 0.00 | 2.69 | 0.02 | .112 | .000 |
| Pythia-70M | refined | 10 | 0.64 | 0.00 | 4.49 | 0.30 | .061 | -.001 |
| Gemma-2 base | penalty $z$, $\lambda$=10 | 100 | 0.77 | 0.00 | 59.55 | 0.53 | – | – |
| Gemma-2 IT | penalty $z$, $\lambda$=10 | 20 | 0.47 | 0.00 | 22.03 | 0.27 | .900 | .010 |
| Gemma-3 base | coordinate | 2 | 0.40 | 0.00 | 0.51 | 0.02 | 1.66 | -.003 |
| Gemma-3 IT | penalty $p$, $\lambda$=0 | 50 | 0.67 | 0.00 | 4.65 | 0.03 | 6.49 | .003 |
| Qwen3-8B | refined | 20 | 0.95† | 0.01 | 1.65 | 0.22 | -.546 | -.007 |

**Data and budgets.** Writer candidates are the MLP channels in every layer up to the feature layer (10,240 for Pythia to 258,048 for Qwen), evaluated on a budget grid from $K = 1$ to 500; consumer candidates are all MLP channels downstream of the edit. Both sides use the three splits defined above. Selected sets are evaluated on held-out data; tuning-split results describe selection (Appendix L). Because the held-out data were consulted while the protocol was developed, Section 4.4 adds an untouched GPT-2 panel. Qwen's confidence probes are steered in both directions, so each steering sign is evaluated separately. The localization and direction-edit tests include random sets of the same size or random edits of the same norm (Appendices A, F, and G).

## 4 Results

We first test whether sparse channels construct the coordinate and carry an edit's effect. We then connect these independently localized sets, test their feature specificity and transfer to new prompts, and ask whether an edit along the direction reproduces the writers' effect.

### 4.1 Sparse writers construct the feature coordinate

The selected writer sets contain 2–100 channels, at most 0.1% of the candidates, and remove 40–95% of the all-candidate reference drop on held-out inputs, compared with at most 1% for equal-size random sets. Their distortion ranges from 0.05% to 6% of the clean residual norm squared (Table 2). On Gemma-3 base, construction keeps rising past the knee at $K = 2$: the 100 top-ranked channels, 0.1% of the candidates, construct 90% of the all-candidate drop at a distortion of only 0.33% (Appendix Figure 3). No tuning-split knee lies at the grid's edge, so the grid captures every selection (Appendix Figure 4).

**Strong coordinate construction need not imply a large behavioral effect.** In the five SAE checkpoint–task settings, ablating the selected writers changes the task endpoint by 0.06–6.5 in each task's endpoint units, against at most 0.010 for random sets. Relative to the all-candidate intervention, these sets reproduce 11–19% of its behavioral effect on GPT-2, Gemma-2 IT, and Gemma-3 base and

Table 3: Held-out absolute restoration (%) at each checkpoint's selected budget; the largest value in each row is in bold. Random is the mean of three random sets of equal size.

| Checkpoint | $K$ | Attribution | 8-step IG | Screened exact | Response | Grad.×act. | Random |
|---|---|---|---|---|---|---|---|
| GPT-2 Small | 50 | 86.40 | **86.81** | 86.40 | 73.81 | -33.12 | -0.46 |
| Pythia-70M | 100 | **58.00** | **58.00** | **58.00** | -0.27 | 15.69 | 3.41 |
| Gemma-2 IT | 100 | 66.41 | **70.31** | 58.59 | 28.91 | 21.88 | -6.25 |
| Gemma-3 base | 200 | 93.80 | **94.03** | 92.94 | 75.70 | 87.63 | 0.37 |
| Gemma-3 IT | 50 | **90.76** | 90.68 | 90.57 | 45.16 | 61.00 | 0.21 |

Table 4: Fraction of the mean signed behavioral effect of the operating-point writer set removed by restoring the frozen consumer set ($M_W$, Eq. (5)) and by three random consumer sets of the same size (range). The last column applies the same test to the coordinate-matched direct edit, where that edit has an effect to mediate. Values above 100% mean that restoration overshoots the clean endpoint.

| Checkpoint | $\lvert S\rvert$ | $\lvert\mathcal{C}\rvert$ | Consumers (%) | Random (%) | Direct edit (%) |
|---|---|---|---|---|---|
| GPT-2 Small | 3 | 50 | 74.0 | -0.2 to -0.1 | 98.8 |
| Pythia-70M | 10 | 100 | 86.9 | 0.2 to 8.2 | 160.7 |
| Gemma-2 IT | 20 | 100 | 87.9 | -1.6 to 1.5 | 75.8 |
| Gemma-3 base | 2 | 200 | 93.2 | -0.8 to 0.3 | 96.4 |
| Gemma-3 IT | 50 | 50 | 64.7 | 0.1 to 0.3 | 72.2 |
| Qwen3-8B | 20 | 5 | 40.1 | -3.9 to 12.4 | – |

59% on Gemma-3 IT: a set can construct most of the coordinate while carrying a smaller share of the behavior, a separation that Section 4.5 examines directly (Appendix Table 11).

**Exact refinement helps, and a cheaper ranking nearly matches it.** On Pythia, exact refinement lowers held-out distortion by 30% relative to the coordinate-ranked set of the same size while raising construction from 0.60 to 0.64. Across checkpoints it wins more matched-budget comparisons than it loses against the coordinate, endpoint, and per-channel penalty rankings (Appendix A.1). A path-integrated ranking, which averages coordinate gradients along the path from the reference to the clean activation, nearly matches it: refinement wins 11 comparisons against it and loses 10, at an estimated 8 to 33 times the cost in forward-equivalent passes.

### 4.2 Sparse consumers carry the effect of direction edits

Attribution-ranked consumers restore 58–94% of the direction edit's absolute effect, against $-6\%$ to $+3\%$ for equal-size random sets (Table 3). The sets are sparse: on Gemma-3 IT, 50 of 82,944 downstream MLP channels (0.06%) restore 91% of the absolute effect. Their signed attenuation is 48–98%, and 109% on Gemma-2 IT, where restoration overshoots (Appendix Table 23).

**Combining edit response and endpoint sensitivity works best.** Among the tested rankings, attribution, IG, and screened exact patching restore the most. Response magnitude, which picks channels that react to the edit whether or not they affect the endpoint, restores 74% of the absolute effect on GPT-2 but none on Pythia; gradient times activation picks endpoint-relevant channels whether or not the edit reaches them, and restoring its GPT-2 set increases the edit's effect ($-33\%$). Attribution combines both signals and exceeds both on every checkpoint (Appendix G). Eight-step IG is within 0.4 percentage points of attribution's absolute restoration on four of the five SAE checkpoints and restores 3.9 points more on Gemma-2 IT, so one gradient usually suffices.

**Consumers recover published task neurons.** On GPT-2, attribution restricted to MLP-10 recovers the published three-neuron Greater-Than set exactly, and the unrestricted ranking gains 24 percentage points of absolute restoration over the eleven published neurons at equal size (Appendix G.2).

Table 5: Mean absolute behavioral change per example for the operating-point writer ablation, the coordinate-matched direct edit at 1, 3, and 10 times its amplitude, and edits of equal norm along ten random directions (mean over directions). The last column is the direct-to-random ratio. Unlike the signed mean effects $\Delta B_W$ of Table 2, these are mean absolute changes.

| Checkpoint | Writer | Direct | $\times 3$ | $\times 10$ | Random | Ratio |
|---|---|---|---|---|---|---|
| GPT-2 Small | .112 | .070 | .306 | 1.15 | .009 | 7.6 |
| Pythia-70M | .145 | .075 | .241 | 1.00 | .013 | 5.7 |
| Gemma-2 IT | 1.02 | .071 | .156 | .382 | .050 | 1.4 |
| Gemma-3 base | 1.66 | 1.99 | 9.60 | 10.4 | .198 | 10.1 |
| Gemma-3 IT | 6.49 | .138 | .505 | 4.78 | .083 | 1.7 |
| Qwen3-8B | .661 | .090 | .123 | .284 | .088 | 1.0 |

### 4.3 Writers act through the localized consumers

**Consumers carry both native and steered effects.** Consumers discovered from steering also transmit native writer effects, although the two sets were localized by different interventions on different inputs and consumer selection never used a writer ablation. Holding the frozen consumer set at its clean activations removes 65–93% of the behavioral effect of ablating the operating-point writer set in the five SAE settings and 40% on Qwen, against $-4\%$ to 12% for random consumer sets of the same size (Table 4; protocol in Appendix H).

The same test removes 72–99% of the coordinate-matched direct edit's effect, and 161% on Pythia, where restoring the consumers overshoots the clean endpoint. The connection persists across alternative Gemma-2 writer sets, with mediation remaining at 88–89%. Pythia's smaller writer effect makes its fraction noisier, and Qwen's 40% varies with the selected set (20–29% at the alternatives; Appendices L and B).

### 4.4 Feature specificity and transfer to new prompts

The consumers could be generic task channels that carry any intervention on the task. We cross consumer sets selected for different features on the same task, then test the fixed GPT-2 sets on new prompts.

**Consumer sets prefer their own feature.** Consumer sets selected for different features on the same task are not interchangeable (Appendix I). On GPT-2 we selected consumers for three further SAE features of the same task and crossed every consumer set with every feature's edit. Averaged across the four features, a feature's own consumer set exceeds the mean of the other three sets in absolute restoration of its edit by 40 percentage points (the own-minus-others contrast). The own set ranks first for three of the four feature edits.

The preference also holds for the native F2448 writer intervention: F2448's own consumers mediate 74% of the effect, 36.7 percentage points more than the mean of the other three sets (Appendix Tables 27 and 28). Each consumer set thus prefers its own feature on top of a substantial shared dependence. On Pythia, two of three features prefer their own consumers, with a mean contrast of 26 percentage points (Appendix J).

**The fixed sets transfer to new prompts.** To check transfer beyond the panels consulted during development, we evaluate the fixed GPT-2 sets on 40 new nouns absent from the checked Greater-Than panels. The fixed sets retain their construction (0.61 against 0.61 on the original panels), writer mediation (75% against 74%), and feature preference (mean contrast 42 percentage points against 40; Appendix K).

### 4.5 Native writers reveal when steering is faithful

The supplied direction is a faithful handle on GPT-2 and Gemma-3 base, where an edit along it approximates the writers' behavioral effect. The edit departs from that effect on the two IT checkpoints and Qwen and is intermediate on Pythia. Across the budget grid (Appendix Table 19),

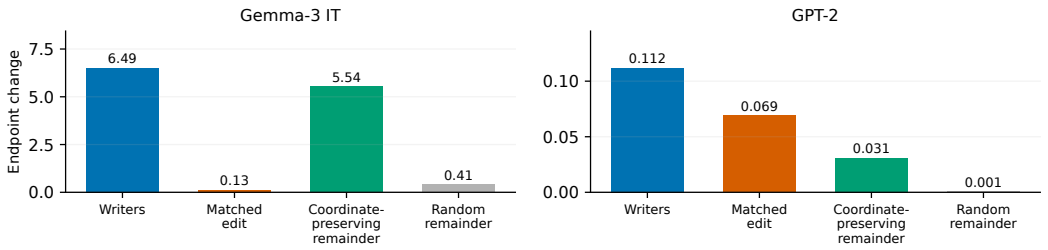



Figure 2: Replaying the writers' residual change on Gemma-3 IT (logits) and GPT-2 (probability difference): the full change, its component along the edit direction (matched edit), the remainder that preserves the coordinate, and an equal-norm random remainder. Bars are separate interventions, not additive shares.

writer and coordinate-matched effects agree within a factor of 2.1 from $K = 3$ upward on GPT-2 and Gemma-3 base, and converge with budget on Pythia.

Example by example (Appendix F.2), the matched edit tracks the ablation on Gemma-3 base and GPT-2, with relative discrepancies of 0.23 and 0.38 and the same sign on 100% and 99% of examples. On the two IT checkpoints it agrees in direction but not in size, and on Pythia and Qwen it shares a sign on 63% and 61% of examples.

**Writers stay strong where matched edits approach random edits.** In mean absolute effect, the matched edit is 5.7–10.1 times stronger than an equal-norm random edit on GPT-2, Pythia, and Gemma-3 base (Table 5). On Gemma-2 IT and Gemma-3 IT it is 1.4–1.7 times stronger, and on Qwen it matches a random edit, while the writers of these three checkpoints change behavior by 7.5–78 times the random edit. There, ten times the matched amplitude recovers 37–74% of the writers' effect.

**The same feature is a much weaker handle in the IT checkpoint.** Holding the feature and prompts fixed, we compare the two Gemma-3 checkpoints on the same 80 held-out rows. At $K = 10$, writer ablation moves the endpoint by 9.03 logits in the base checkpoint and 4.13 in the IT checkpoint, while the matched edit moves it by 5.44 and 0.11, respectively: the edit's effect differs by a factor of about fifty between the two checkpoints on the same feature, prompts, and rows. On every row the matched edit reproduces the writers' coordinate to within 0.5% of its change (Appendix F.3), so at the matched amplitude moving along the SAE direction does not reproduce the change the writers make.

**On Gemma-3 IT, the coordinate-preserving remainder alone reproduces 85% of the writer effect.** To examine what the matched edit misses, we split the writers' residual change into a component along the edit direction and a remainder that preserves the coordinate at the target token, and replay each (Figure 2; Appendix F.1). On Gemma-3 IT, the part that leaves the coordinate unchanged moves the endpoint by 5.54 of the writers' 6.49 logits, 14 times an equal-norm random control, while the matched part moves it by 0.13. Applied alone or added to the remainder, the matched component changes the mean endpoint by at most 0.95 logits (15% of the full writer effect). The components interact, so these are separate intervention effects, not mediated shares (Appendix F.1).

## 5 Related Work

**Residual features and steering.** Linear probes measure which properties can be decoded from an activation (Alain & Bengio, 2016), and the linear representation hypothesis proposes that concepts correspond to directions in activation space (Park et al., 2023). Steering vectors and representation engineering edit these directions to control behavior (Turner et al., 2023; Zou et al., 2023; Rimsky et al., 2024; Jiang et al., 2026), and SAEs decompose activations into sparse, interpretable directions (Elhage et al., 2022; Bricken et al., 2023; Huben et al., 2024; Lieberum et al., 2024; Rajamanoharan et al., 2024). Evaluations of SAE latents as steering handles compare them with supervised directions

(Arad et al., 2025; Wu et al., 2025). We localize the native channels that write and read such directions and use native writer interventions as a reference for evaluating edits along them.

**Circuit discovery and attribution.** Circuit analysis finds task subgraphs of attention heads and MLPs (Conmy et al., 2023; Wang et al., 2022; Hanna et al., 2023), and the Mechanistic Interpretability Benchmark (MIB) evaluates methods that localize them (Mueller et al., 2025). Attribution patching approximates intervention effects with gradients (Syed et al., 2024); AtP* addresses failure modes of this approximation (Kramár et al., 2024), and EAP-IG uses integrated gradients (Hanna et al., 2024). Edge pruning learns sparse edge masks (Bhaskar et al., 2024). Sparse feature circuits use SAE latents as circuit nodes (Marks et al., 2025). Transcoders approximate MLP computations with sparse learned features (Dunefsky et al., 2024), and circuit tracing uses cross-layer transcoders to construct attribution graphs (Ameisen et al., 2025). Feed-forward layers act as key–value memories whose down-projections write residual directions (Geva et al., 2021; 2022), which knowledge localization and model editing exploit (Dai et al., 2022; Meng et al., 2022). These methods search for task components or build new features; we start from a supplied direction, map its writers and consumers to native channels, and test them by exact intervention.

**Causal intervention methodology.** Causal mediation analysis and interchange interventions test whether a representation carries an effect (Vig et al., 2020; Geiger et al., 2024; Wu et al., 2023). Subspace patching can mislead (Makelov et al., 2024), patching results depend on the metric and baseline (Zhang & Nanda, 2024), and later layers can compensate for ablations (McGrath et al., 2023). Channel localization intervenes on native activations, while consumer discovery and the handle comparison edit the subspace deliberately, against random controls. Encoding need not imply influence (Sharma & Le, 2026); our direct edits test a direction's influence against the writers.

## 6 Discussion

**Implications for steering.** Connecting the maps shows that an edit along the direction at the writers' coordinate change and the writers themselves act through the same consumers yet can produce very different behavioral effects. Native writers therefore offer a reference for testing whether a steering direction captures the model's computation.

**Practical guidance.** Inexpensive attribution rankings locate candidate channels, exact interventions evaluate the selected sets, and native writers and random edits test a direction before it is used as a handle. The path-integrated ranking is a cheap alternative to exact refinement, which adds a tuning-split guarantee; for consumers, single-gradient attribution usually suffices (Sections 4.1 and 4.2).

**Limitations and scope.** The maps cover MLP channels. Replacing attention outputs moves the coordinate several times more than replacing the MLP pool on Pythia and Qwen (Appendix C), so $C$ measures the MLP contribution rather than the whole feature; attention writers are not localized here. The cases are published or task-grounded features rather than a sample of all features. The original held-out panels were consulted during protocol development; the new GPT-2 panel confirms that the fixed sets transfer. On four checkpoints the operating point depends on the candidate methods (Appendix Table 9). Cross-feature tests cover two checkpoints and one native writer set per checkpoint. Identifying what the coordinate-preserving change encodes is the next step.

## 7 Conclusion

The native MLP channels that construct a supplied direction's coordinate and those through which it reaches behavior are both sparse: as little as 0.1% of the candidate channels construct 90% of the MLP contribution to a feature, and 0.06% of downstream channels restore 91% of a steering edit's effect. The writers act through the consumers that carry steering, and consumer sets prefer their own feature. On Gemma-3 IT, the matched edit reproduces little of the writers' effect, while the part of the writers' change that leaves the coordinate unchanged reproduces 85% of it. Feature-direction localization thus turns a steering direction into a testable account of native computation: where the feature is written, how it reaches behavior, and when editing it reproduces the writers' effect.

### AI USE STATEMENT

In this work, we used generative AI tools to write the templates and nouns of the Gemma-3 dataset (Appendix M.1), to implement methods, and to design and give feedback on experiments and methodology. We did not use generative AI tools for the other tasks that require disclosure. Additionally, we used generative AI tools to write and edit code, create figures and tables, search and summarize literature, and draft and edit the paper. We have reviewed all AI-assisted work: every number quoted in the paper is recomputed from saved per-example records by a validation script, and the authors read and revised all text. We did not use generative AI tools to generate references. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

Feature localization can support model auditing and targeted behavioral interventions. The same interventions can alter useful model behavior, so results should be interpreted within the reported tasks and intervention settings. The study involves no human subjects and uses existing model and SAE releases and the evaluation materials described in Appendix M.

### REPRODUCIBILITY STATEMENT

Appendices A–M describe the interventions, data splits, selection rules, and hyperparameters, and name the public model and SAE releases each case uses. Intervention results are generated from per-example records rather than from summary statistics, so each can be recomputed or reaggregated; the subset-transfer diagnostic is computed from stored attributions. Scripts regenerate every table and figure and recompute every number quoted in the paper from these records. The code and records will be released.

## REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, et al. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 6(16318-16352):1, 2025.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. Saes are good for steering–if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10252–10270, 2025.

Aryaman Arora, Zhengxuan Wu, Jacob Steinhardt, and Sarah Schwettmann. Language model circuits are sparse in the neuron basis. *arXiv preprint arXiv:2601.22594*, 2026.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. *Advances in Neural Information Processing Systems*, 37:18506–18534, 2024.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations*, volume 2025, pp. 77965–78000, 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 5484–5495, 2021.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 30–45, 2022.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060, 2023.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. *arXiv preprint arXiv:2403.17806*, 2024.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Nick Jiang, Isaac Kauvar, and Jack Lindsey. The value axis: Language models encode whether they're on the right track. *arXiv preprint arXiv:2606.17056*, 2026.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, 2024.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, volume 2024, pp. 26486–26515, 2024.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, volume 2025, pp. 23888–23923, 2025.

Callum McDougall, Arthur Conmy, János Kramár, Tom Lieberum, Senthooran Rajamanoharan, and Neel Nanda. Gemma scope 2-technical paper. *Google DeepMind Blog*, 2025.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, et al. Mib: A mechanistic interpretability benchmark. *arXiv preprint arXiv:2504.13151*, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Or David Shafran, Atticus Geiger, and Mor Geva. Constructing interpretable features from compositional neuron groups. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42326–42348, 2026.

Aarushi Sharma and Phong Le. Encoding without influence: Dissociating demographic representation from causal effect in large language mod-els. *Transactions on Machine Learning Research*, 2026.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in neural information processing systems*, 36:78205–78226, 2023.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.