

How Far Do SAE Features Persist Across LLM Layers? An Empirical Study with an FFN Pruning Application

Vaishakh Sreekanth Menon

Independent Researcher
vaishakhsmn@gmail.com

Asif Ekbal

Indian Institute of Technology Patna
Patna, India
asif@iitp.ac.in

Abstract

Sparse autoencoders (SAEs) are typically applied at the single layer they were trained on. We ask whether a pretrained SAE’s per-prompt feature activations persist when the same SAE is applied to residuals at deeper layers of the underlying language model, and whether the resulting cross-layer signal is strong enough to drive per-prompt FFN neuron pruning without any training or calibration data. Testing three SAE families on three roughly 8B-scale language models, we find adjacent-layer carry-through is high across families (0.72–0.82 at $L_{\text{src}}+1$) and its rate of decay differs qualitatively across sparsification mechanisms, with JumpReLU still activating half the features 17 layers downstream while TopK degrades within about five. We propose **FARN** (Feature and Anchor-based Removal of Neurons), a training-free per-prompt FFN pruner that scores each downstream neuron on the SAE’s decoder columns weighted by the current prompt’s active features, plus a prompt-independent anchor from the decoder bias. On six multiple-choice benchmarks, keeping roughly 17–20% of FFN neurons per layer, FARN closes the gap to an activation oracle (a baseline that peeks at the true downstream activations to pick the top-firing neurons) to within 1.5 percentage points and outperforms random, magnitude, and calibrated static-activation baselines on every model.

1 Introduction

It has been shown that in modern LLMs, only a small fraction of feed-forward neurons contribute meaningfully to each FFN layer’s output for any given token (Liu et al., 2023; Zhang et al., 2022). Subsequently various pruning methods have exploited this, differing mainly in how the important subset is identified. Some train a predictor that forecasts activations from the current residual (Liu et al., 2023; Akhauri et al., 2024) or apply

activation-magnitude statistics calibrated on a held-out set (Lee et al., 2024). Others prune statically using activation-weighted magnitude and Hessian-based importance scores (Sun et al., 2024; Frantar and Alistarh, 2023).

Sparse autoencoders (SAEs) have emerged as a standard tool for identifying interpretable features in the activations of transformer language models (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2024). Each SAE is trained on the residuals at a single layer, so its indices stand for features whose meaning comes from that layer. Encoding a residual gives us a value for each index, and the few that are nonzero tell us which semantic features are present at that layer and how strongly. Separately, prior work has shown that many layers of pretrained LLMs are functionally redundant (Gromov et al., 2025; Men et al., 2025), an observation that has been leveraged to train sparse autoencoders shared across groups of adjacent layers (Ghilardi et al., 2025). Motivated by these two threads, we ask what happens when an SAE trained at layer L is applied to residuals at deeper layers $L' > L$. We hypothesize that the features active at L stay active at L' , and that this leaves enough signal to guide FFN neuron pruning there. We call this feature persistence, and it is a claim about the persistence of which indices fire and how strongly, not about the semantic meaning of an index, since that meaning is tied to the layer the SAE was trained on. In the remainder of the paper, when we refer to a feature we mean the index the SAE activates, not the semantic feature it corresponds to. A method that re-uses an off-the-shelf SAE would remain calibration-free and training-free, and every pruning decision it makes would be traceable to SAE features that fired at the source layer. We test this hypothesis on three modern ~ 8 B-scale base language models paired with three publicly released SAEs from different sparsification variants (TopK, modified-TopK, and JumpReLU; details in §4.1),

and propose FARN, a novel FFN pruning method that operationalizes the findings. Our contributions are:

- We empirically characterize SAE feature persistence across layers: Applying a single SAE at layers deeper than its training layer, we find adjacent-layer feature carry-through (the fraction of features active at layer L that remain active when the same SAE is applied to a deeper layer) is high across the three SAE families we test (0.72–0.82 at the adjacent layer) and decays with depth at a rate that differs qualitatively across sparsification mechanisms. For instance, JumpReLU retains half the features 17 layers downstream while TopK-based SAEs retain the same fraction only 4–5 layers downstream.
- We propose **FARN** (Feature and Anchor-based Removal of Neurons), an SAE-feature-based FFN pruning technique: FARN scores each downstream neuron on features activated in previous layers together with a feature-invariant anchor term. On six multiple-choice benchmarks, keeping roughly 17–20% of FFN neurons per layer, FARN comes within 1.5 percentage points of an oracle baseline and is competitive with weight-only and calibration-based baselines.

2 Related Work

Our work sits at the intersection of three lines of research: cross-layer behavior of SAE features, SAEs as downstream signals for practical tasks, and FFN pruning. This section places our contribution against each of these.

2.1 Cross-layer behavior of SAE features

Whether SAE features written into the residual stream at one layer remain meaningful at deeper layers has been approached in a few ways. Some work jointly trains a single SAE on residuals from multiple layers: Group-SAE (Ghilardi et al., 2025) clusters layers by residual angular distance and fits one SAE per cluster, while MLSAE (Lawson et al., 2025) trains a single TopK SAE on residuals from every Pythia layer. A second line introduces new architectures with per-layer decoder heads: Sparse Crosscoders (Lindsey et al., 2024) decompose residuals from several layers into

shared features with dedicated per-layer reconstructions. Crosscoders model cross-layer feature sharing by design, whereas our work asks a question of whether an unmodified single-layer SAE, one that was never trained with cross-layer objectives, carries usable cross-layer signal. A third line uses off-the-shelf per-layer SAEs and learns a permutation-based alignment between them. SAE Match (Balogansky et al., 2025) does this on Gemma Scope and reports feature persistence over roughly five adjacent layers. Our setup differs in that we apply a single unmodified SAE across all downstream layers with no matching.

2.2 SAEs as downstream signals

This body of work uses SAEs not as objects of study but as *signals* for downstream tasks: interpretable, monosemantic features (Bricken et al., 2023) usable for concept steering (Templeton et al., 2024), and sparse feature circuits for causal analysis and editing (Marks et al., 2025), all treating SAE features as actionable handles on model behavior. This established pattern of SAE-as-signal also motivates our approach to ask whether SAE features are meaningful enough to *prune* FFN compute per prompt.

2.3 FFN pruning

Per-prompt FFN pruning is naturally motivated by ReLU architectures where activation sparsity is exact. Modern SwiGLU and GeGLU FFNs do not produce exact zeros, and recent work restores ReLU-style sparsity through activation replacement (Mirzadeh et al., 2024) or explicit sparsity regularization (Song et al., 2025). We take modern SwiGLU/GeGLU FFNs as given and treat per-prompt pruning as top- K magnitude selection over their long-tailed activation distributions.

Existing methods in this regime fall broadly into three families with three mask granularities. *Learned predictors* (Liu et al., 2023; Akhauri et al., 2024) train a small forecaster that recomputes the mask at every decoding step from the current residual. *Calibrated statistics* (Lee et al., 2024) similarly threshold per token using a magnitude profile computed on a held-out corpus. *Static importance methods* (Sun et al., 2024; Frantar and Alistarh, 2023) compute a single mask offline using activation-weighted magnitude and Hessian-based importance scores, then reuse it across all inputs. Between the per-token and static extremes lies a middle ground where the mask is fixed at prefill

and reused throughout decoding, giving per-prompt sparsity without per-token overhead; this is where FARN sits.

Most directly comparable to our setting is the training-free, calibration-free subniche. GLASS (Sattarifar et al., 2026) combines a local activation-magnitude signal with a global prior computed from self-generated activation statistics via null-prompt stimulation, requiring no held-out calibration corpus, and produces per-prompt masks without either training a predictor or curating calibration data. We share this training-free, calibration-free regime, but our pruning signal has a different origin. We score each downstream neuron by how strongly the SAE features that fired in earlier layers (via their decoder directions) align with that neuron’s gate/up projections, plus a feature-invariant anchor term from the decoder bias. Because the score depends on which features fired, the resulting masks are per-prompt, and each pruning decision is traceable to specific, individually interpretable SAE features at the source layer, a property that activation-statistic priors cannot provide.

3 FARN

FARN assigns each FFN neuron at every downstream layer $L' > L_{\text{src}}$ two scores, a prompt-conditional score derived from the SAE’s decoder columns and a prompt-independent anchor score derived from the SAE’s decoder bias. At inference, the K -neuron budget is partitioned between the two scores by a coefficient $\beta \in [0, 1]$. The top $\lfloor \beta K \rfloor$ neurons by anchor score and the top $K - \lfloor \beta K \rfloor$ (disjoint) neurons by prompt-conditional score are selected, and their union forms the kept set for layer L' . Figure 1 shows the overall pipeline.

Notation. We consider decoder-only transformers whose feed-forward blocks use a gated linear unit variant (Shazeer, 2020), element-wise multiplying a nonlinearly-activated gate against a linear up-projection. At each block L , the FFN computes

$$\text{FFN}_L(r) = W_{\text{down}}^L (\sigma(W_{\text{gate}}^L r) \odot (W_{\text{up}}^L r)), \quad (1)$$

where $W_{\text{gate}}^L, W_{\text{up}}^L \in R^{d_{\text{ffn}} \times d_{\text{model}}}$, $W_{\text{down}}^L \in R^{d_{\text{model}} \times d_{\text{ffn}}}$, and σ is the gate activation (SiLU in the SwiGLU variant used by Qwen and Llama, GELU in the GeGLU variant used by Gemma). Each neuron j owns one row of W_{gate}^L and one of

W_{up}^L , and one column of W_{down}^L . We call the scalar

$$h^L[j] = \sigma(W_{\text{gate}}^L[j, :] r) \cdot (W_{\text{up}}^L[j, :] r) \quad (2)$$

the *intermediate output* of neuron j at layer L : it is the scalar coefficient that gets multiplied against the j -th output column of W_{down}^L and added into the residual stream. *Pruning* neuron j means forcing $h^L[j] = 0$, which skips its contribution to the FFN’s output entirely and, in practice, lets an inference engine avoid computing the corresponding rows of W_{gate}^L , W_{up}^L and column of W_{down}^L for that token.

A pretrained SAE at layer L_{src} decomposes the residual as

$$r_{L_{\text{src}}} \approx b_{\text{dec}} + \sum_{k=1}^{d_{\text{feat}}} a_k w_k, \quad (3)$$

where $w_k \equiv W_{\text{dec}}[:, k]$ is the residual-space direction of feature k , $W_{\text{dec}} \in R^{d_{\text{model}} \times d_{\text{feat}}}$ collects all such directions, and $b_{\text{dec}} \in R^{d_{\text{model}}}$ is a learned bias. Only a small subset $S \subset [d_{\text{feat}}]$ of the coefficients a_k is non-zero (typically $|S| \approx 50\text{--}250$ depending on family); we write $a_S \in R^{|S|}$ for their values. For readability we also abbreviate the gate and up read directions of neuron j at layer L as $g_j^L \equiv W_{\text{gate}}^L[j, :]$ and $u_j^L \equiv W_{\text{up}}^L[j, :]$.

The alignment matrix. We want a per-neuron score that attributes each SAE feature k to the neurons at downstream layer L' most responsible for realizing its contribution. Feature-attribution methods answer questions of this shape either by integrating gradients along an input path (Sundarajan et al., 2017) or by linearizing causal interventions (Nanda, 2023; Marks et al., 2025), both of which require running the model on real inputs. A closely related *weight-space* line applies fixed projections between component matrices to attribute residual-space directions to downstream computation. The logit lens (nostalgebraist, 2020) reads any intermediate residual through the unembedding, with subsequent refinements such as the tuned lens (Belrose et al., 2025), and this discipline has since been used to characterize both individual FFN neurons (Gurnee et al., 2024) and SAE features (Bloom and Lin, 2024) via alignments with W_U . We adopt the same weight-space discipline but change the target. Rather than projecting through the unembedding to read what a component predicts, we project SAE decoder columns

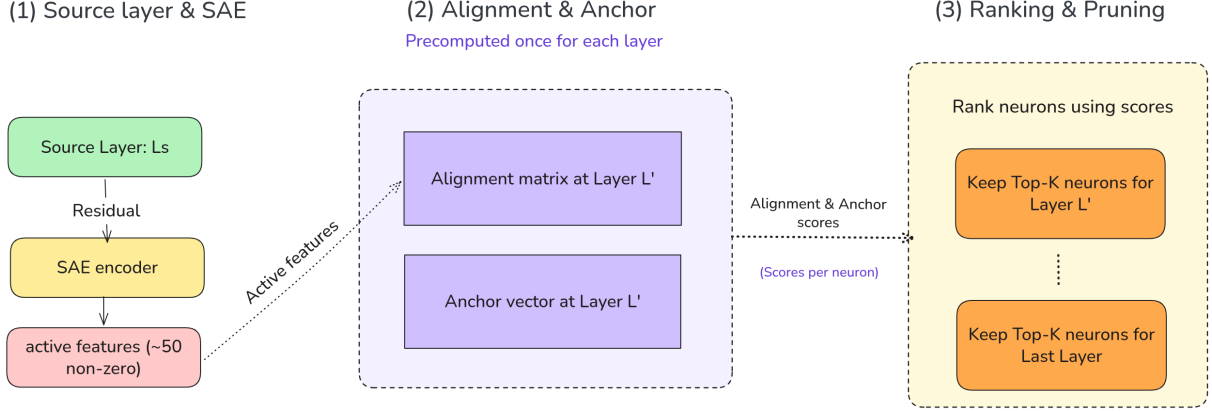


Figure 1: FARN inference pipeline. (1) The residual at source layer L_{src} is encoded by a SAE, yielding a sparse set of active features. (2) For every downstream layer L' , an alignment matrix $A^{L'}$ and an anchor vector $c^{L'}$ are precomputed. (3) Alignment & anchor scores rank neurons at downstream layers, and the top- K are kept.

through the downstream FFN’s gate and up matrices to read which neurons will fire on a given feature. If feature k is active with strength a_k , its first-order contribution to neuron j ’s gate pre-activation is $a_k (g_j^{L'} \cdot w_k)$ and to its up projection is $a_k (u_j^{L'} \cdot w_k)$. Since $h^{L'}[j]$ multiplies σ of the gate against the up projection, feature k ’s sensitivity to the intermediate scales, to first order, as $|g_j^{L'} \cdot w_k| \cdot |u_j^{L'} \cdot w_k|$, and a neuron responds strongly to feature k only when both its gate and up rows are aligned with w_k . Factoring out the prompt-dependent a_k , the purely geometric per-feature attribution is the alignment matrix

$$A^{L'}[j, k] = |g_j^{L'} \cdot w_k| \cdot |u_j^{L'} \cdot w_k|, \quad (4)$$

a data-free, per-feature attribution score derivable from weight geometry alone. The prompt-conditional score used at inference reweights these entries by the active a_k values, defined in the selection rule below. Absolute values collapse the sign, treating gate and up symmetrically as a linearized alignment proxy that does not model σ ’s nonlinearity or possible suppression of features that align negatively with the gate.

The anchor score. The decoder bias b_{dec} is a fixed direction in residual space that the SAE learned as the baseline residual it reconstructs to when no feature is active. Applying the same product-of-absolutes construction gives a prompt-independent per-neuron score:

$$c^{L'}[j] = |g_j^{L'} \cdot b_{\text{dec}}| \cdot |u_j^{L'} \cdot b_{\text{dec}}|. \quad (5)$$

$c^{L'}$ ranks neurons by how strongly they respond to the SAE’s baseline residual direction alone, without any prompt-specific feature.

The ranking. At inference on a new prompt, we forward the model up to layer L_{src} and encode the residual through the SAE to obtain the active-feature vector a_S . Given a target sparsity budget K (neurons kept per layer) and a mixing coefficient $\beta \in [0, 1]$, FARN scores every FFN neuron at every downstream layer L' and selects

$$s^{L'}[j] = \sum_{k \in S} A^{L'}[j, k] a_k, \quad (6)$$

$$\mathcal{K}_{\text{anchor}}^{L'} = \text{top-}(\lfloor \beta K \rfloor) \text{ neurons by } c^{L'} \quad (7)$$

$$\mathcal{K}_{\text{sae}}^{L'} = \text{top-}(K - \lfloor \beta K \rfloor) \text{ neurons by } s^{L'} \quad (8)$$

The two sets are chosen disjoint, with the SAE-scored neurons drawn from the neurons not already in $\mathcal{K}_{\text{anchor}}^{L'}$. The final kept set is $\mathcal{K}^{L'} = \mathcal{K}_{\text{anchor}}^{L'} \cup \mathcal{K}_{\text{sae}}^{L'}$; all other neurons at L' have their intermediate output set to zero. The anchor half of $\mathcal{K}^{L'}$ acts as a prompt-independent set of universally-live neurons that respond to the SAE’s baseline direction; the SAE half is a prompt-conditional set of neurons selected specifically for the features currently active. The mixing coefficient β trades off between the two, and its optimum is studied empirically in Section 5.2.

4 Experimental Setup

4.1 Models and SAEs

We evaluate on three modern base LLMs, each paired with a publicly released SAE from a different sparsification family: Qwen3-8B (Yang et al., 2025) with a TopK SAE from QwenScope (Gao et al., 2025; Deng et al., 2026), Llama-3.1-8B (Grattafiori et al., 2024) with a modified-TopK SAE from Llama Scope (He et al., 2024),

and Gemma-2-9b (Gemma Team et al., 2024) with a JumpReLU SAE from Gemma Scope (Rajamanoharan et al., 2024; Lieberum et al., 2024). Qwen and Llama use the SwiGLU FFN variant; Gemma uses GeGLU. Every SAE is applied at layer $L_{\text{src}}=24$ and used without any retraining, fine-tuning, or per-layer adaptation. We choose $L_{\text{src}}=24$ because a shallower source layer would prune the early FFN blocks, where a mask error carries through more of the network that follows, and a deeper one would leave too few FFN layers below it for pruning to have any effect. At layer 24, 17 of Gemma’s 42 FFN layers remain available to prune, 11 of Qwen’s 36, and 7 of Llama’s 32, which is 40%, 31%, and 22% of each network. We use the same layer for all three models so that the family comparison is not confounded with source depth. We do not sweep L_{src} , so its effect on FARN’s accuracy remains open. Persistence and FARN scoring experiments (§4.2, §4.3) sample prompts from wikitext-103 (Merity et al., 2017), truncated to at most 512 tokens each.

4.2 Question 1: Feature persistence

We measure whether the SAE’s active feature set at L_{src} survives when the same SAE is applied at deeper layers. (We compare activation patterns, not semantic feature meanings.)

Setup. For each model, we forward $n=500$ wikitext prompts and, at each layer $L' \geq L_{\text{src}}$, capture the residual at the last 8 non-pad token positions, yielding 4000 (prompt, position) samples per (model, layer). We encode each captured residual through the layer- L_{src} SAE, obtaining the sparse activation vector $a_{L'}$. Persistence is the extent to which $a_{L'}$ resembles $a_{L_{\text{src}}}$.

Metrics and rationale. We report three quantities per (model, layer):

- **Carry rate** = $|S_{\text{src}} \cap S_{L'}|/|S_{\text{src}}|$, the fraction of features active at L_{src} that remain active at L' .
- **Pearson(a) on S_{src}** , for features active at L_{src} , correlation of their activation strengths at L_{src} versus L' . Answers whether persisting features preserve their magnitude, not just their identity; FARN’s score $s^{L'} = \sum_k A^{L'}[j, k] a_k$ is linear in the a_k values, so magnitude drift attenuates the score directly.

- **FVU at L'** , fraction of variance unexplained when the layer- L_{src} SAE reconstructs $r_{L'}$. Measures whether the SAE’s decoder basis still covers the deeper residual as a whole, independent of which specific features fire. Reconstruction fidelity is a separate property from identification fidelity, and comparing the two lets us diagnose whether an SAE has stopped reconstructing but is still identifying (or vice versa).

Carry and Pearson describe whether the SAE fires the same features at the same magnitudes as at L_{src} ; FVU describes whether the SAE reconstructs the residual well enough for the model to use it downstream. FARN’s per-prompt score depends on the former (the SAE encoder’s activation pattern) and does not use reconstructions at all, so the two sets of metrics report on independent axes of SAE behaviour.

4.3 Question 2: FARN’s pruning signal and downstream performance

If features persist, we still need to check two things about FARN itself: does its score actually predict which neurons fire, and does pruning on it preserve model behaviour on real tasks? We test the first geometrically (Q2a) and the second functionally (Q2b).

Q2a: does the score predict actual firing? Using the same 4000 (prompt, position) samples per (model, layer), we compute the FARN score $s^{L'}[j]$ for every FFN neuron at every downstream layer, and separately the actual intermediate output magnitude $|h^{L'}[j]|$ from the clean forward pass. Magnitude is the pruning-relevant target because pruning zeros a neuron’s contribution to the residual. We measure the agreement between the two rankings using two metrics. Spearman rank correlation (ρ) over all d_{ffn} neurons quantifies how well FARN’s predicted ranking matches the actual magnitude ranking. Top- K overlap, defined as $|\text{top-}K(s^{L'}) \cap \text{top-}K(|h^{L'}|)|/K$, is the fraction of the K neurons FARN would keep that are among the K hardest-firing; this is the pruning-relevant quantity we emphasize, since ρ can be inflated by the dense tail of small-magnitude neurons.

Q2a ablations. We further sweep two knobs whose defaults FARN needs to justify. The **K -sweep** varies $K \in \{500, 1000, 2500, 5000, 7500\}$ at $\beta=0.5$, and reports top- K overlap at each K ;

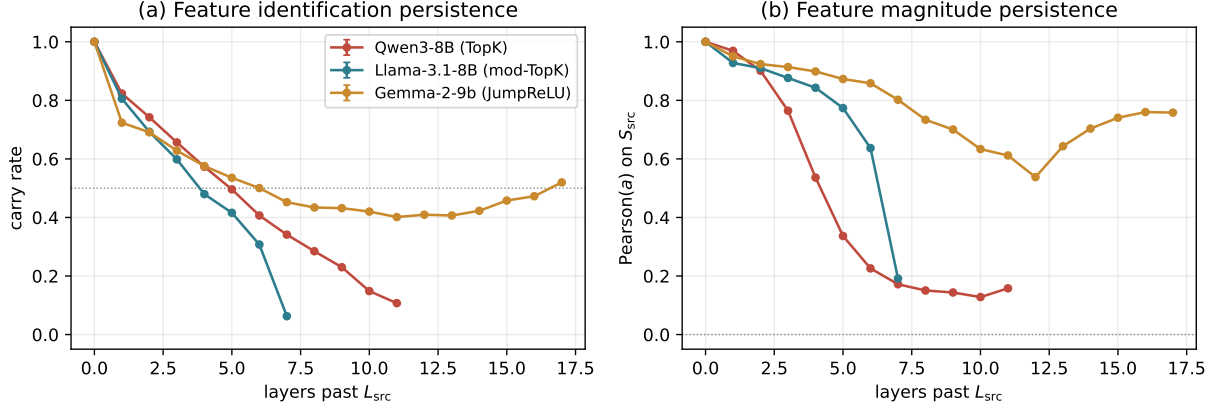


Figure 2: SAE feature persistence across depth for three sparsification families. (a) Carry rate: fraction of features active at L_{src} that remain active at L' . (b) Pearson correlation of activation magnitudes over S_{src} . JumpReLU (Gemma) sustains identification and magnitude far into depth; TopK (Qwen) decays sharply after ~ 5 layers; modified-TopK (Llama) shows identity decay with preserved magnitudes.

this shows whether the score is sharpest at small K , i.e. whether the neurons the score ranks highest really are the ones most likely to fire. The β -sweep varies $\beta \in \{0.25, 0.5, 0.75\}$ at $K=2500$, and reports top- K overlap for the mixed selection; this probes whether the optimal budget split between anchor and SAE score is empirically consistent with the FARN default.

Q2b: does pruning preserve task performance?

We evaluate FARN pruning on six standard English multiple-choice benchmarks: MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC-Easy, ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), and PIQA (Bisk et al., 2020). These are a widely used evaluation suite for base LLMs and cover reasoning, commonsense, and factual recall. We use the lm-eval-harness (Gao et al., 2024) with a limit of 300 examples per task and zero-shot prompting. For each model we compare FARN ($\beta=0.5$, $K=2500$) against four baselines, all evaluated at the same budget $K=2500$, plus the unpruned upper bound:

- **No pruning:** the full model, kept as an upper bound on accuracy.
- **Random pruning:** at every layer, keep a random subset of K neurons and zero the rest. Prompt-independent; a lower bound.
- **Magnitude pruning:** for each neuron j at layer L' , compute the combined weight magnitude $\|W_{gate}^{L'}[j, :]\|_2 \cdot \|W_{up}^{L'}[j, :]\|_2$; keep

the K largest. Weight-only and prompt-independent.

- **Static activation-frequency (calibrated):** compute the mean absolute intermediate output $|h^{L'}[j]|$ for each neuron over a calibration set of held-out prompts; keep the K most-frequently-active neurons. Prompt-independent, but uses calibration data.
- **Per-prompt activation (activation oracle):** at inference, run the FFN in full and rank neurons by their actual intermediate output $|h^{L'}[j]|$ on the current prompt; keep the top K . We call this an *oracle* because it has direct access to the exact quantity FARN predicts from proxies, so no method that must commit to a mask before running the FFN can beat it at reproducing $FFN(L')$'s per-layer output. It is not a competitive pruner because it must run the full gate and up projections to compute $|h^{L'}[j]|$ and only saves work on W_{down} , and its per-layer optimality does not directly translate to downstream task accuracy, since top- K by $|h|$ ignores the W_{down} column norms and interactions with subsequent layers.

We do not directly compare against DejaVu- or ShadowLLM-style trained predictors. Those methods train a per-layer forecaster on labelled activations from the base model and therefore inhabit a strictly stronger data regime than ours, so a same-budget comparison would not be an apples-to-apples test of the pruning signal itself.

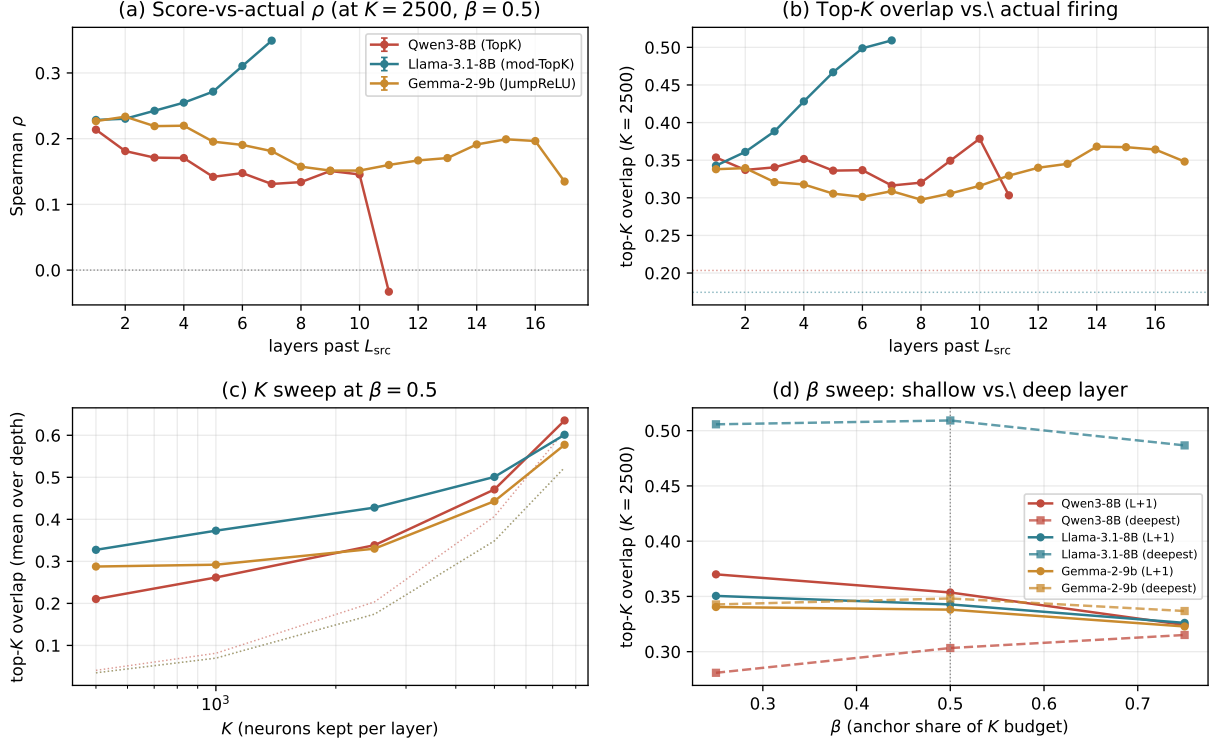


Figure 3: FARN pruning signal quality across three models at $L_{\text{src}}=24$. (a, b) Vary downstream layer L' at $K=2500, \beta=0.5$. (a) Spearman ρ between predicted and actual neuron ranking. (b) Top- K overlap with the top- K firing neurons (dashed = random baseline K/d_{fin}). (c) K -sweep at $\beta=0.5$, averaged over downstream layers. (d) β -sweep at $K=2500$, at two downstream layers per model (solid = $L_{\text{src}}+1$, dashed = deepest measured). Optimum β increases with depth in every family.

5 Results

5.1 Feature persistence

Figure 2 summarises persistence across depth for all three SAE families. **Adjacent-layer carry-through is high across families:** at $L_{\text{src}}+1$, carry rate ranges from 0.72 (Gemma) to 0.82 (Qwen), with Llama at 0.81; Pearson(a) is above 0.93 in all three cases. The SAE at L_{src} transfers to the very next layer with almost no loss, regardless of sparsification mechanism.

Extended-depth carry rates diverge qualitatively across families. JumpReLU (Gemma) retains half the features 17 layers past L_{src} ; TopK (Qwen) drops below half within 5; modified-TopK (Llama) shows the fastest identity decay but preserves activation magnitudes (Pearson stays above 0.6 through $L+6$). These trajectories bound the depth over which FARN’s SAE-derived pruning signal remains useful, and imply that work needing cross-layer feature attribution gets further with a JumpReLU checkpoint than with a comparable TopK one.

Reconstruction fidelity decays faster than

identification. FVU stays near its in-domain L_{src} value one layer past L_{src} across all families, but grows with depth. Qwen’s FVU stays between 0.4 and 0.7 across intermediate depths before jumping to 0.87 at its final layer, Llama’s grows monotonically from 0.55 to 2.30 over 7 layers, and Gemma’s blows up from 1.45 in-domain to 24.05 at $L=41$ (Appendix C). By the deepest layers on each model, identification signal outlasts reconstruction fidelity.

On Gemma we also observe that carry rate and FVU move somewhat independently across depth: on the three JumpReLU L_0 checkpoints we test (34, 61, 114), identification stays roughly L_0 -invariant while reconstruction FVU grows with depth and worsens sharply with lower L_0 . We do not draw a general claim from this since we do not sweep L_0 on TopK or modified-TopK families; the corresponding curves are in Appendix B. We also measured ΔCE , the increase in next-token cross-entropy when the SAE reconstruction is spliced into the forward pass at L' , across all three families; the picture matches FVU, with functional reconstruction loss nontrivial from $L_{\text{src}}+1$ onward and growing with depth in a family-dependent way

(Appendix D).

5.2 FARN’s pruning signal

Figure 3 combines the four measurements FARN’s pruning signal must pass. Panels (a) and (b) vary depth at the default $K=2500$, $\beta=0.5$. **Spearman ρ is positive at every layer of Llama and Gemma and at all but the deepest layer of Qwen**, with values ranging from -0.03 (Qwen’s final layer, where the SAE feature signal collapses) up to 0.35 (Llama’s deepest layer). **Top- K overlap sits comfortably above the random baseline** (0.20 for Qwen; 0.17 for Llama and Gemma) at every depth, with headline overlap values roughly 0.30 – 0.51 . On Llama, both metrics counterintuitively *rise* with depth despite persistence collapsing (carry rate reaches 0.06 at $L=31$); we suspect this reflects the deep-layer FFN concentrating its firing on a smaller set of neurons, making the top- K ranking easier to hit for any reasonable score.

Panels (c) and (d) vary K and β . Because FARN operates at downstream layers $L' > L_{\text{src}}$, both sweeps have to pick specific L' choices; panel (c) averages the K -sweep over all downstream layers, and panel (d) traces the β -sweep at two representative L' per model, the near-source layer $L_{\text{src}}+1$ (solid, $L=25$) and each model’s deepest measured layer (dashed). The **K -sweep** shows the signal is sharpest at the top of the ranking; at $K=500$ (top 4% of neurons), overlap is several times above the random baseline on every model, and as K grows toward d_{ffn} overlap trivially approaches 1, so the informative regime is small K , exactly the pruning-relevant regime. The **β -sweep** reveals a pattern common to all three families, **optimal β increases with depth**, from SAE-heavy near L_{src} (best at $\beta=0.25$) to anchor-heavier at the deepest layers, consistent with the SAE feature signal weakening at depth and the prompt-independent anchor taking over. The effect is most dramatic on Qwen (optimum shifts from $\beta=0.25$ at $L=25$ to $\beta=0.75$ at $L=35$), and mildest on Gemma, whose JumpReLU signal barely weakens. The default $\beta=0.5$ is a compromise that never trails the depth-optimal choice by more than a few points. Comparing $\beta=0.5$ directly against the SAE-only ($\beta=0$) and anchor-only ($\beta=1$) endpoints across all depths further shows that the mix protects against both failure modes: the SAE-only endpoint collapses toward random at Qwen’s deepest layers, and the anchor-only endpoint consistently lags near L_{src} on every model (Appendix A).

5.3 Downstream pruning

Table 1 reports zero-shot accuracy on six MC benchmarks with $K=2500$ neurons kept per FFN layer (roughly 17–20% retention). **FARN matches the activation oracle within 1.5 percentage points (pp) across all three models**: the gap is 1.5 pp on Qwen, -0.5 pp on Llama (FARN slightly exceeds the oracle), and 1.3 pp on Gemma. Since the oracle uses the model’s own downstream activations at each pruned layer to pick the true top- K firing neurons, it is the natural ceiling for reproducing the FFN’s per-layer output. It is not a strict upper bound on downstream accuracy, as the -0.5 pp Llama result illustrates: top- K by $|h|$ is not equivalent to top- K by contribution to the residual downstream. Closing the gap to 1.5 pp with a purely weight-space, training-free score is our headline outcome.

FARN also beats the training-free / calibration-based baselines by wider margins. Against random pruning, the improvement is 10.7–22.2 pp; against magnitude-based static pruning, 2.5–23.2 pp; against calibrated static-activation pruning, 0.9–4.2 pp. The gap over calibrated static activation is narrowest on Llama, where the model’s activation patterns are stable enough that a corpus-averaged prior is competitive, and widest on Qwen. On Gemma, magnitude-based pruning collapses (accuracy drops from 0.724 to 0.403), but FARN’s SAE-feature pruning preserves 88% of unpruned accuracy.

Calibration-dependent baselines can also suffer under domain shift, since their pruning signal is computed on one distribution and applied on another. FARN has nothing to calibrate. The alignment matrix and anchor come from weights and SAE decoders, and per-prompt selection depends only on which SAE features fire at L_{src} on the current prompt, so this failure mode does not apply by construction.

6 Conclusion

We show that a single SAE trained at one layer of a modern LLM continues to fire a substantial fraction of the same features when applied to residuals at layers well past its training point, with the rate of decay differing qualitatively across sparsification mechanisms. This cross-layer signal is strong enough to drive a training-free per-prompt FFN pruner (FARN) whose pruning decisions come within 1.5 percentage points of an activation oracle

Method	Qwen3-8B		Llama-3.1-8B		Gemma-2-9b	
	Acc	Δ	Acc	Δ	Acc	Δ
No pruning (upper bound)	0.677	—	0.677	—	0.724	—
Random	0.500	−17.7	0.534	−14.3	0.413	−31.1
Magnitude	0.545	−13.2	0.616	−6.1	0.403	−32.1
Static activation (calibrated)	0.575	−10.3	0.632	−4.5	0.596	−12.8
Per-prompt activation (oracle)	0.632	−4.5	0.636	−4.1	0.648	−7.6
FARN, SAE-only ($\beta=0$)	0.608	−6.9	0.629	−4.8	0.610	−11.4
FARN (ours, $\beta=0.5$)	0.617	−6.0	0.641	−3.6	0.635	−8.9

Table 1: Zero-shot accuracy on six MC benchmarks (MMLU, HellaSwag, ARC-Easy, ARC-Challenge, WinoGrande, PIQA; 300 examples each), averaged over tasks. Kept-neuron budget $K=2500$. Accuracy drop (Δ) is in percentage points versus the no-pruning upper bound. FARN closes the gap to the activation oracle to within 1.5 pp on all three models and beats every training-free or calibration-based baseline. The SAE-only row ($\beta=0$) shows that the anchor contributes 0.9–2.5 pp on top of the SAE-alignment signal.

and outperform magnitude, random, and calibrated static-activation baselines. Every FARN pruning decision remains traceable to specific SAE features at the source layer, a property that activation-statistic priors and trained predictors do not afford.

Limitations

We evaluate three ~ 8 B-scale base models spanning three SAE families; scaling behaviour and family-dependent trajectories at 70B and above are open. Downstream numbers are reported at a single sparsity budget ($K=2500$) and a single source layer ($L_{\text{src}}=24$). We do sweep K for the top- K overlap in Figure 3(c), but the corresponding downstream accuracy-versus- K curve and an L_{src} sweep are natural follow-ups that more aggressive sparsity or different source layers could reshape. Persistence is measured through activation patterns, not the semantic content of the features that fire, so whether persisting features also persist in meaning is a separate question we do not test.

References

- Yash Akhauri, Ahmed F. AbouElhamayed, Jordan Dotzel, Zhiru Zhang, Alexander M. Rush, Safeen Huda, and Mohamed S. Abdelfattah. 2024. [ShadowLLM: Predictor-based contextual sparsity for large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19154–19167, Miami, Florida, USA. Association for Computational Linguistics.
- Nikita Balagansky, Ian Maksimov, and Daniil Gavrilov. 2025. [Mechanistic permutability: Match features across layers](#). In *The Thirteenth International Conference on Learning Representations*.
- Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2025. [Eliciting latent predictions from transformers with the tuned lens](#). *arXiv preprint arXiv:2303.08112*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. [PIQA: Reasoning about physical commonsense in natural language](#). In *AAAI Conference on Artificial Intelligence*.
- Joseph Bloom and Johnny Lin. 2024. [Understanding SAE features with the logit lens](#). *AI Alignment Forum*.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, 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, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Boyi Deng, Xu Wang, Yaoning Wang, Yu Wan, Yubo Ma, Baosong Yang, Haoran Wei, Jialong Tang, Huan Lin, Ruize Gao, Tianhao Li, Qian Cao, Xuancheng Ren, Xiaodong Deng, An Yang, Fei Huang, Dayiheng Liu, and Jingren Zhou. 2026. [Qwen-scope: Turning sparse features into development tools for large language models](#). *arXiv preprint arXiv:2605.11887*.
- Elias Frantar and Dan Alistarh. 2023. Sparsegpt: massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan

- Leike, and Jeffrey Wu. 2025. [Scaling and evaluating sparse autoencoders](#). In *The Thirteenth International Conference on Learning Representations*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. [Gemma 2: Improving open language models at a practical size](#). *arXiv preprint arXiv:2408.00118*.
- Davide Ghilardi, Federico Belotti, Marco Molinari, Tao Ma, and Matteo Palmonari. 2025. [Group-SAE: Efficient training of sparse autoencoders for large language models via layer groups](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18657–18677, Suzhou, China. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Gloriosio, and Dan Roberts. 2025. [The unreasonable ineffectiveness of the deeper layers](#). In *The Thirteenth International Conference on Learning Representations*.
- Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. [Universal neurons in GPT2 language models](#). *Transactions on Machine Learning Research*.
- Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. [Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders](#). *arXiv preprint arXiv:2410.20526*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. [Sparse autoencoders find highly interpretable features in language models](#). In *The Twelfth International Conference on Learning Representations*.
- Tim Lawson, Lucy Farnik, Conor Houghton, and Laurence Aitchison. 2025. [Residual stream analysis with multi-layer SAEs](#). In *The Thirteenth International Conference on Learning Representations*.
- Donghyun Lee, Jaeyong Lee, Genghan Zhang, Mo Tiwari, and Azalia Mirhoseini. 2024. [CATS: Context-aware thresholding for sparsity in large language models](#). In *First Conference on Language Modeling*.
- Tom Lieberum, Senthooan Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. [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*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.
- Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. 2024. [Sparse crosscoders for cross-layer features and model diffing](#). *Transformer Circuits Thread*.
- Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. 2023. [Deja vu: contextual sparsity for efficient llms at inference time](#). In *Proceedings of the 40th International Conference on Machine Learning, ICLR’23*. JMLR.org.
- Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. [Sparse feature circuits: Discovering and editing interpretable causal graphs in language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. [ShortGPT: Layers in large language models are more redundant than you expect](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, Vienna, Austria. Association for Computational Linguistics.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Seyed Iman Mirzadeh, Keivan Alizadeh-Vahid, Sachin Mehta, Carlo C del Mundo, Oncel Tuzel, Golnoosh

- Samei, Mohammad Rastegari, and Mehrdad Farajtabar. 2024. [ReLU strikes back: Exploiting activation sparsity in large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Neel Nanda. 2023. [Attribution patching: Activation patching at industrial scale](#). Blog post.
- nostalgebraist. 2020. [Interpreting GPT: The logit lens](#). *AI Alignment Forum*.
- Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. [Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders](#). *arXiv preprint arXiv:2407.14435*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. [Winogrande: an adversarial winograd schema challenge at scale](#). *Commun. ACM*, 64(9):99–106.
- Amirmohsen Sattarifard, Sepehr Lavasani, Kunlin Zhang, Amirhossein Rajabpour, Hanlin Xu, Fengyu Sun, Negar Hassanpour, and Chao Gao. 2026. [Glass: Global-local aggregation for inference-time sparsification of llms](#). *arXiv preprint arXiv:2508.14302*.
- Noam Shazeer. 2020. [Glu variants improve transformer](#). *arXiv preprint arXiv:2002.05202*.
- Chenyang Song, Xu Han, Zhengyan Zhang, Shengding Hu, Xiyu Shi, Kuai Li, Chen Chen, Zhiyuan Liu, Guangli Li, Tao Yang, and Maosong Sun. 2025. [ProSparse: Introducing and enhancing intrinsic activation sparsity within large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2626–2644, Abu Dhabi, UAE. Association for Computational Linguistics.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. [A simple and effective pruning approach for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3319–3328. JMLR.org.
- 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, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. [Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet](#). *Transformer Circuits Thread*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.
- Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. [MoEification: Transformer feed-forward layers are mixtures of experts](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, Dublin, Ireland. Association for Computational Linguistics.

A FARN component ablation

Figure 4 isolates the two components of FARN’s pruning signal, showing top- K overlap at $K=2500$ for the SAE-only endpoint ($\beta=0$), the anchor-only endpoint ($\beta=1$), and the default mixed selection ($\beta=0.5$) at every downstream layer for each model. The mixed selection is either the winner or a safe near-winner at every depth. The SAE-only endpoint collapses toward the random baseline at Qwen’s deepest layers, and the anchor-only endpoint consistently lags near L_{src} across all three models. The mix therefore protects against both failure modes: it inherits the SAE signal where features are still identifiable, and falls back to the prompt-independent anchor when they are not.

B Gemma-2-9b L_0 sweep: carry rate and FVU

Figure 5 shows carry rate and FVU on Gemma-2-9b with the pretrained Gemma Scope SAE at $L_{\text{src}}=24$, for three JumpReLU L_0 checkpoints ($L_0 \in \{34, 61, 114\}$). Carry rate is nearly L_0 -invariant; FVU separates cleanly and grows with depth. We do not sweep L_0 on the other SAE families.

C Reconstruction FVU across depth

Table 2 reports the fraction of variance unexplained (FVU) when the layer- L_{src} SAE reconstructs the residual at each downstream layer L' , at four depth checkpoints per model. The in-domain FVU at L_{src} differs across families, reflecting the different SAE architectures rather than any cross-layer effect. From $L_{\text{src}}+1$ onward, FVU grows with depth in a

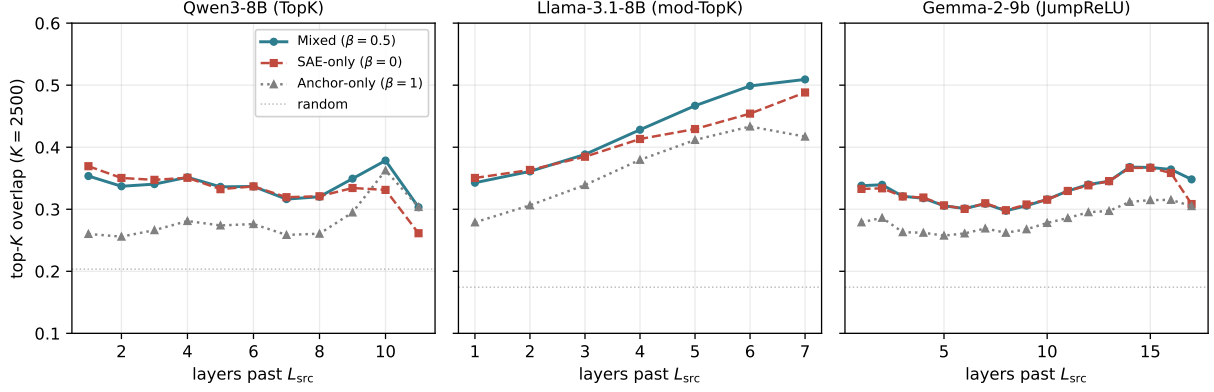


Figure 4: Top- K overlap at $K=2500$ for the three FARN variants across downstream layers, one panel per model. Mixed ($\beta=0.5$, solid teal) is the FARN default. SAE-only ($\beta=0$, dashed red) and anchor-only ($\beta=1$, dotted grey) are the two endpoints. Dotted horizontal line marks the random baseline K/d_{ffn} .

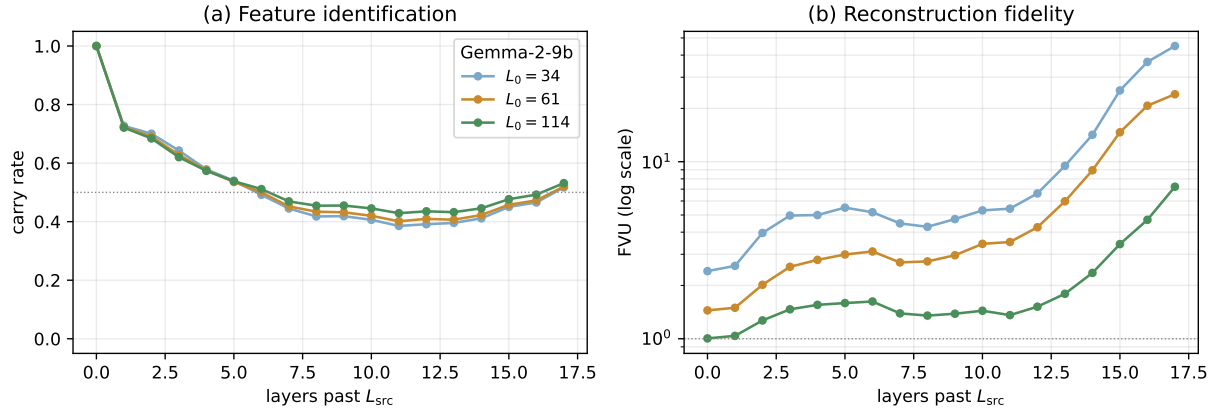


Figure 5: Gemma-2-9b feature persistence and reconstruction fidelity across three JumpReLU L_0 checkpoints (34, 61, 114). (a) Carry rate is nearly identical across the three L_0 settings. (b) FVU separates cleanly, growing with depth and worsening at lower L_0 .

family-dependent way, with Gemma’s rise the most dramatic.

Model	24	25	30	Deepest
Qwen3-8B	0.70	0.62	0.52	0.87
Llama-3.1-8B	0.43	0.55	1.37	2.30
Gemma-2-9b	1.45	1.50	3.11	24.05

Table 2: FVU at four depths per model, with the SAE taken from $L_{\text{src}}=24$. Column headings are layer indices. The deepest measured layer is 35 for Qwen, 31 for Llama and 41 for Gemma. Standard errors are ≤ 0.15 and are omitted for readability.

D Reconstruction ΔCE across depth

As a functional check on reconstruction fidelity we also measured ΔCE , the increase in next-token cross-entropy loss when the layer- L_{src} SAE’s reconstruction of the residual is spliced back into the forward pass at each downstream L' , on the first

Model	25	30	Deepest
Qwen3-8B	+1.83	+1.86	+11.24
Llama-3.1-8B	+1.64	+3.97	+8.06
Gemma-2-9b	+2.49	+1.86	+4.95

Table 3: ΔCE (nats) at three depths per model, with the SAE taken from $L_{\text{src}}=24$. Column headings are layer indices. The deepest measured layer is 35 for Qwen, 31 for Llama and 41 for Gemma. Standard errors are ≤ 0.09 nats and are omitted for readability.

100 wikitext prompts. Table 3 reports ΔCE at three depth offsets per model.

ΔCE at $L_{\text{src}}+1$ is already 1.6–2.5 nats across every family, so the SAE reconstruction is not functionally interchangeable with the true residual even one layer past L_{src} . Depth trajectories track FVU, with Gemma rising modestly, Llama monotonically, and Qwen staying flat until its final layers, where it blows up.