TopK LM: Towards Language Models with Intrinsic SAE-level Interpretability

Anonymous ACL submission

Abstract

Sparse autoencoders (SAEs) are widely used to interpret language models (LMs), but because they are trained post hoc, they inherit reconstruction error and feature instability, and raise ambiguity about whether missing features are absent from the model or not learned by the SAE. We propose an alternative in which SAE-like sparsity is built directly into the model by applying TopK gating to residual-stream hidden states after transformer blocks. We ask whether building sparsity directly into the model can yield an intrinsically interpretable language model, and if so, at what capability cost. Under a quantitative evaluation, we find a clear performance-interpretability trade-off: quality recovers at higher k, while interpretability peaks at an intermediate sparsity level and then declines again on the higher-performance side. Qualitatively, individual TopK neurons support direct interpretation and steering, and geometric analyses suggest that TopK induces a structured reorganization of dense representations, not simple representational degradation. Overall, these results position TopK LMs as a concrete architecture for intrinsic interpretability research, where this trade-off can be measured and analyzed directly.

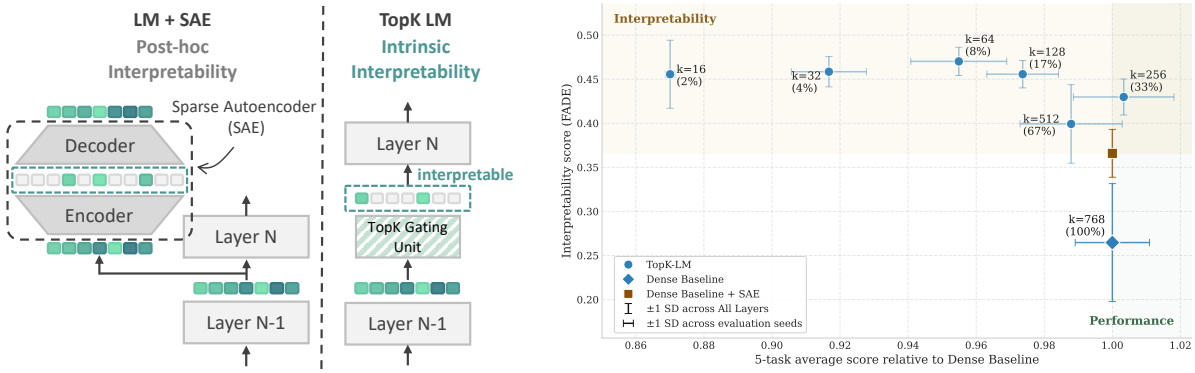
1 Introduction

Sparse autoencoders (SAEs; Sharkey et al., 2023; Cunningham et al., 2023; Templeton et al., 2024), in particular TopK SAEs (Gao et al., 2024), have become an important tool for interpreting the internal representations of language models (LMs) (Lieberum et al., 2024; He et al., 2024; Karvonen et al., 2025; Shu et al., 2025). However, they suffer from several shortcomings that diminish their utility and internal validity. Because SAEs are trained post-hoc and remain lossy, they introduce a trade-off between sparsity and reconstruction fidelity (Rajamanoharan et al., 2024; Gao et al., 2024). This makes it harder to tell whether a missing feature is absent from the LM or simply not discovered by the SAE, and the problem is compounded by sensitivity to architecture choice, random seed, and checkpoint (Leask et al., 2025; Paulo and Belrose, 2025; Xu et al., 2025; Inaba et al., 2025). These limitations become especially costly when the goal is to trace how features form during training, because the SAE interpretation method must itself be retrained and then aligned across checkpoints before feature-level comparisons are possible.

In this work, our goal is to devise an LM architecture that enjoys the interpretability benefits of SAEs without relying on post-hoc training. Specifically, we achieve SAE-like sparsity by applying the TopK activation function to the residual stream after transformer blocks, as illustrated in Figure 1a and defined formally in Section 2.2, while leaving the attention and MLP computations within each block unchanged. In the main comparison, we apply this gating in all transformer blocks and compare the resulting TopK LMs with dense baselines and their post-hoc SAE features.

We ask whether building sparsity into the model can produce an intrinsically interpretable LM. Figure 1b shows that it can, though only through a non-monotonic trade-off: broader capability is strongest at higher k, whereas the clearest interpretability gains appear at intermediate sparsity. This improvement is not free, because the stronger TopK LMs require additional optimization to recover quality. At the same time, once sparsity is built directly into the model, the same latent dimensions can be inspected directly, followed across checkpoints, and intervened on without aligning a separate feature space after training. That makes TopK LMs a concrete setting for intrinsic interpretability research, where individual dimensions can be read as specific concepts, single-dimension interventions can steer generation, and the same dimensions can be traced throughout training.

In summary, this work makes the following con-



(a) Post-hoc vs. Intrinsic Interpretability

(b) Performance-Interpretability Trade-off

Figure 1: **TopK LM architecture and the main performance-interpretability trade-off.** (a) contrasts TopK LM, which builds sparsity directly into the model by applying TopK gating to the residual stream after each transformer block, with a dense baseline LM interpreted with a separately trained post-hoc SAE. (b) shows the 12-layer trade-off across TopK LMs with different k values. The horizontal axis is the five-task average score relative to the dense baseline, and the vertical axis is the interpretability score. As k increases, task performance recovers, but interpretability peaks at intermediate sparsity rather than at the highest-performance end of the curve.

tributions:

- We propose TopK LM, a transformer-based LM architecture in which TopK is applied to the residual stream after transformer blocks, yielding sparse dimensions in the model itself.
- We quantify how capability and interpretability change across TopK LMs with different k values, and show that the most favorable trade-off appears at intermediate sparsity, not at either end of the sweep.
- We show that TopK LMs learn sparse dimensions that can be interpreted directly, steered causally, and traced across checkpoints, and that geometric analysis helps explain the resulting trade-off.

2 Method and Evaluation

2.1 Motivation

A useful way to view TopK is as an intervention on representation geometry. Prior work on superposition suggests that when many sparse features must share limited dimensions, models can encode them as approximately orthogonal directions in a dense space (Elhage et al., 2022). Such representations are efficient, but they need not align with the coordinate axes that we inspect as neurons. TopK counteracts this by enforcing exact per-token sparsity and competition across dimensions: after an arbitrary rotation, a different set of coordinates would be selected, so rotated bases are no longer equivalent. This creates pressure for features to align with specific dimensions, closely related to the idea of a privileged basis (Elhage et al., 2021).

We apply this pressure to the residual stream because it is the transformer's shared cross-layer communication space. If the goal is to encourage an interpretable basis, sparsifying that shared channel is more direct than sparsifying a transient subcomputation inside one block. Our claim is therefore modest but meaningful: TopK creates a cleaner setting in which more axis-aligned, stable, and intervention-friendly features can emerge.

2.2 Architecture

In this section we describe the TopK LM architecture. The key intervention is to apply TopK gating to the residual stream after the standard block update in each transformer block, while leaving the attention and MLP computations unchanged, as illustrated in Figure 2. This design preserves dense computation inside each block while sparsifying only the state passed forward to the next layer. Geometrically, the sparsity constraint therefore acts on the LM's shared representational bandwidth as a whole, which is why it can encourage feature alignment without forcing attention and MLP subcomputations themselves to become sparse.

TopK activation Let $\tilde{\boldsymbol{h}}_\ell \in \mathbb{R}^{d_{\text{model}}}$ denote the residual-stream hidden state after the standard transformer-block update at layer ℓ, and let $\mathcal{L}_{\text{topk}}$ denote the set of layers in which TopK is applied.

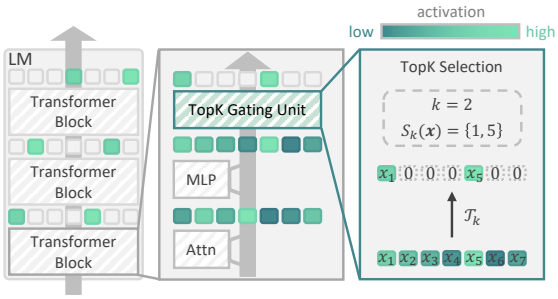


Figure 2: TopK LM block design. In selected transformer blocks, the attention and MLP computations remain unchanged, and a TopK gating unit is applied to the residual-stream hidden state after the block update. The unit first retains the top-k coordinates and then applies a nonlinearity, yielding a sparse hidden state for the next block.

The block output is then

$$\boldsymbol{h}_{\ell+1} = \begin{cases} \mathcal{T}_k(\tilde{\boldsymbol{h}}_\ell) & \ell \in \mathcal{L}_{\text{topk}} \\ \tilde{\boldsymbol{h}}_\ell & \text{otherwise.} \end{cases} \tag{1}$$

Here k is a hyperparameter that controls the sparsity level by selecting how many coordinates are retained. As k varies, the same architecture ranges from the dense case ($k = d_{\text{model}}$) to strictly sparse residual states ($k < d_{\text{model}}$). For any $\boldsymbol{x} = (x_1, \ldots, x_{d_{\text{model}}}) \in \mathbb{R}^{d_{\text{model}}}$, let $S_k(\boldsymbol{x}) \subseteq \{1, \ldots, d_{\text{model}}\}$ denote the index set of the selected top-k coordinates of $\boldsymbol{x}$, with ties broken arbitrarily. The TopK activation operator $\mathcal{T}_k \colon \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}$ is defined component-wise by

$$[\mathcal{T}_k(\boldsymbol{x})]_j = \begin{cases} x_j & j \in S_k(\boldsymbol{x}) \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, \ldots, d_{\text{model}}, \tag{2}$$

In this identity-postact form, $k = d_{\text{model}}$ recovers the dense baseline exactly because $\mathcal{T}_k$ becomes the identity map.

2.3 Pre-training Settings

We train TopK and dense decoder-only transformer LMs in the nanochat framework (Karpathy, 2025), comparing TopK variants with dense baselines throughout. All dense and TopK models in the main comparisons are trained from scratch and do not rely on pretrained initialization. Across these comparisons, models share the same tokenizer, pretraining data, width recipe, model size, optimizer setup, and evaluation procedure; the controlled differences are the TopK budget k and whether TopK gating is present. Our main results use TopK-only gating with no residual add-back, no additional post-TopK nonlinearity, and no annealing.

For the main comparison, we allow sparser TopK LMs to train for more steps so that active units receive more comparable optimization. This choice is motivated by prior evidence from sparse optimization that reusing dense training schedules without adjustment can under-train sparse models (Kuznedelev et al., 2024). Concretely, we use a simple heuristic in which the number of training steps scales with d_{model}/k: if the dense-width point ($k = d_{\text{model}}$) uses T steps, then a TopK model with width k is trained for $T \cdot (d_{\text{model}}/k)$ steps. We treat this as a pragmatic comparison heuristic, not an exact equality claim. This longer-horizon TopK protocol is the one used for the 12-layer and 24-layer main comparisons. As a robustness check, we also verified in appendix-side IsoFLOP-matched comparisons that the same broad non-monotonic tradeoff remains visible, with interpretability again peaking at an intermediate sparse point; see Tables 10 and 13. Tokenizer and data details, optimization constants, and hardware details are deferred to Section C.

2.4 Evaluation

Task Evaluation Settings We evaluate model quality with validation loss together with a small representative task set. The five tasks are chosen to cover distinct capability types while avoiding floor-heavy benchmarks in the small-model regime, and their simple mean serves as the compact task-level summary used throughout the main text. We also retain a wider benchmark suite for broader context, but full task lists, prompting details, and scoring rules are deferred to Section C.1.

Interpretability Evaluation Settings For dense baselines, we evaluate sparse concepts in separately trained SAE latents, whereas for TopK LMs we evaluate the sparse dimensions directly. Recent work has started to standardize automated interpretability evaluation and benchmarking for sparse representations (Karvonen et al., 2025; Paulo et al., 2025; Wu et al., 2025; Puri et al., 2025). In this paper, we use FADE (Puri et al., 2025) as the primary automated interpretability metric, and in the main analysis we use the all-layer mean of its main summary score, where higher values indicate stronger feature-description alignment. Independent FADE denotes the standard model-independent use of

Table 1: Results for the 12-layer trade-off comparison across k. Sparser models are trained longer ($k = 768, 512, 256, 128, 64, 32, 16 \mapsto 2\text{k}, 3\text{k}, 6\text{k}, 12\text{k}, 24\text{k}, 48\text{k}, 96\text{k}$ steps). We report validation loss (↓), five representative downstream tasks and their mean (↑), and independent FADE (↑). Downstream scores are averaged over evaluation seeds $42, 43, 44$. The $k = 768$+SAE row uses the same model as $k = 768$, but evaluates FADE on post-hoc SAE latents rather than on the model's native TopK basis.

	Language Modeling	Downstream Tasks						Interpretability
Model	Val. Loss ↓	ARC-e ↑	PIQA ↑	HellaSwag ↑	BBH-CS ↑	CoQA ↑	Avg. ↑	FADE ↑
$k = 768$ (100%)	2.954	56.27	68.07	34.67	**45.67**	15.47	44.03	0.2647
$k = 768$ + SAE	–	–	–	–	–	–	–	0.3659
$k = 512$ (66.7%)	2.968	55.53	68.07	**35.27**	43.40	15.20	43.49	0.3993
$k = 256$ (33.3%)	**2.953**	**57.53**	**69.00**	34.27	44.40	**15.67**	**44.17**	0.4298
$k = 128$ (16.7%)	2.976	56.00	67.93	34.07	41.27	15.07	42.87	0.4557
$k = 64$ (8.3%)	3.019	53.07	67.93	33.13	44.07	12.00	42.04	**0.4702**
$k = 32$ (4.2%)	3.153	53.93	65.67	31.07	40.27	10.87	40.36	0.4585
$k = 16$ (2.1%)	3.412	46.20	63.13	30.67	42.67	8.87	38.31	0.4556

FADE: each model is evaluated on its own target latent set, with no cross-model matching, so it serves as the paper's standalone interpretability measure for each model. Semantically matched FADE is our comparison-specific evaluation mode built on top of the same FADE pipeline: for each comparison between a dense baseline and a TopK LM, we first construct a semantically aligned latent subset and then evaluate FADE only on that subset, so it serves as a more fine-grained dense–TopK comparison and not as an independent model score. The underlying FADE scoring is unchanged; the difference lies only in which latents are evaluated. We use semantically matched FADE as a robustness check instead of the primary ranking. Further details of SAE training and FADE evaluation are deferred to Sections D.1 and D.2.

3 Main Results: Performance-Interpretability Trade-off

This section gives the main quantitative result of the paper. We focus first on a 12-layer full-layer TopK sweep evaluated in the same way across models. Within this comparison, TopK traces out a clear trade-off surface: larger budgets recover language-model quality, but the strongest feature readability appears at intermediate sparsity, not at the highest-performance end.

3.1 Main Results in a 12-Layer Model

We compare seven full-layer TopK models with $k \in \{16, 32, 64, 128, 256, 512, 768\}$ in a 12-layer setting, summarized in Table 1. Across this series, model quality recovers as k moves out of the very sparse regime and is strongest in the high-k region, while interpretability peaks at intermediate sparsity instead of following the same direction. This pattern remains visible in the all-layer average and does not depend on a hand-picked subset of layers.

The best quality point in the completed series is $k = 256$, with validation loss 2.9534, task average 44.17, and all-layer mean FADE 0.4298. The near-dense points $k = 768$ and $k = 512$ remain strong on broad quality, but their all-layer mean FADE values fall to 0.2647 and 0.3993. By contrast, $k = 64$ is the clearest interpretability point: its validation loss is worse (3.0191) and its task average is lower (42.04), but it attains the strongest all-layer mean FADE in the series at 0.4702. The main empirical result is therefore not a capability-preserving gain from sparsity, but a clear performance-interpretability trade-off with the most favorable interpretability point at intermediate sparsity.

3.2 24-Layer Scale-up

We next test whether the same trade-off persists in a deeper 24-layer model. Across the completed sweep, the same broad pattern remains visible: capability is strongest at higher k, while interpretability peaks deeper in the sparse regime. The main difference from the 12-layer comparison is that the 24-layer sweep uses a shorter base training horizon.

Table 2 summarizes the broad performance picture. Within the completed set, $k = 128$ is strongest on capability, with the best validation loss (2.6934) and the best representative-task average (49.52). By contrast, the strongest independent FADE score appears at $k = 32$, where validation loss worsens to 2.9323 and task average drops to

Table 2: Results for the 24-layer scale-up comparison across k. Sparser models are trained longer ($k = 1536, 768, 512, 256, 128, 64, 32 \mapsto 1\text{k}, 2\text{k}, 3\text{k}, 6\text{k}, 12\text{k}, 24\text{k}, 48\text{k}$ steps). We report validation loss (↓), five representative downstream tasks and their mean (↑), and independent FADE as the all-layer mean (↑). As in Table 1, the $k = 1536$+SAE row uses the same model as $k = 1536$ but evaluates FADE on post-hoc SAE latents.

	Language Modeling	Downstream Tasks						Interpretability
Model	Val. Loss ↓	ARC-e ↑	PIQA ↑	HellaSwag ↑	BBH-CS ↑	CoQA ↑	Avg. ↑	FADE ↑
$k = 1536$ (100%)	2.789	63.40	72.40	40.40	**46.60**	15.60	47.15	0.2336
$k = 1536$ + SAE	–	–	–	–	–	–	–	0.3882
$k = 768$ (50.0%)	2.753	63.60	73.60	41.40	42.40	17.40	47.41	0.4369
$k = 512$ (33.3%)	2.743	62.60	72.60	42.40	42.80	18.80	47.16	0.4291
$k = 256$ (16.7%)	2.715	65.20	73.00	45.00	42.40	15.60	46.28	0.4484
$k = 128$ (8.3%)	**2.693**	**65.80**	**75.60**	**48.20**	42.80	**22.20**	**49.52**	0.4566
$k = 64$ (4.2%)	2.765	64.60	69.20	44.60	43.80	18.40	47.83	0.4639
$k = 32$ (2.1%)	2.932	60.20	70.20	40.40	45.20	14.60	45.85	**0.4771**

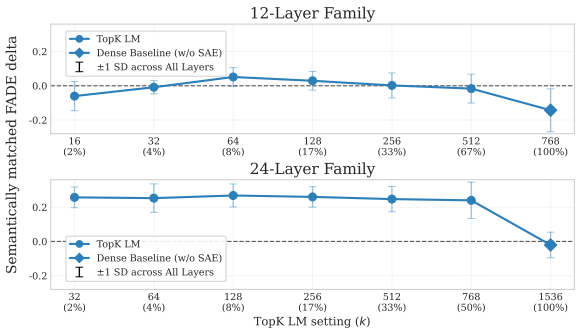


Figure 3: Semantically matched FADE in the 12-layer and 24-layer families. Positive values indicate an advantage for the TopK LM over the dense baseline with SAE on the semantically aligned subset. The 12-layer family shows a matched advantage concentrated around intermediate k, while the 24-layer family remains positive across the sparse sweep.

45.85. Interpretability rises sharply away from the dense-width model, from 0.2336 at $k = 1536$ to 0.4566 at $k = 128$, 0.4639 at $k = 64$, and 0.4771 at $k = 32$. The dense-width model also becomes more readable under a post-hoc SAE control, reaching 0.3882, but still remaining below the strongest TopK points. We therefore read the 24-layer results as a scale-up confirmation of the same overall trade-off pattern: $k = 128$ is the strongest capability point, while the best independent FADE value appears at $k = 32$.

3.3 A Fairer Interpretability Comparison

To make the dense–TopK comparison more controlled, we also evaluate both models on a semantically aligned latent subset. Semantically matched FADE is reported as a fairness-oriented robustness check, not as a replacement for the main independent FADE ranking. The visual summary is given in Figure 3, while the exact values are deferred to Appendix E.2.

Even under this more semantically controlled comparison, the same broad trade-off pattern remains visible at both scales. In the 12-layer sweep, the favorable intermediate-sparsity region remains favorable under semantically matched FADE, while the dense-width model evaluated directly in its native basis remains worse than the matched SAE anchor. In the 24-layer scale-up check, the pattern is even clearer: all completed TopK points beat the matched dense-width SAE anchor on the aligned subset.

Overall, even under the stricter semantically matched evaluation that complements independent FADE, the strongest TopK LMs remain favorable relative to the SAE-based anchors at both scales.

3.4 Discussion of the Trade-off

The task-level losses are category-specific in both the 12-layer main comparison and the 24-layer scale-up check. Across both scales, the largest gaps tend to appear on world-knowledge and language-understanding tasks such as ARC-e, HellaSwag, and CoQA, whereas more symbolic behavior is comparatively preserved on BBH CS Algorithms. This is clearest in the deeper model: relative to $k = 128$, the smaller-budget $k = 64$ point is worse on PIQA (75.6 → 69.2), HellaSwag (48.2 → 44.6), and CoQA (22.2 → 18.4), while BBH CS Algorithms remains comparatively stable (42.8 → 43.8). Taken together, these patterns suggest that the present bottleneck is the loss of distributed semantic capacity under hard, uniform all-layer TopK rather than sparsity in the abstract. The geometry analysis in Section 4 supports the

same reading: the TopK models are better understood as structured sparse reparameterizations than as generic failures.

These comparisons also carry a real optimization cost, since the stronger sparse points are trained longer than the dense-width baseline in the main setup. In the present design, recovering dense-like behavior requires additional training while yielding a more readable sparse basis. This in turn points to a natural next step: relaxing the current all-layer TopK constraint through less uniform sparse designs, such as layer-selective placement, larger TopK budgets in semantically demanding layers, or softer mechanisms like dense-to-sparse annealing and hybrid dense–sparse placement. Appendix Section B collects these exploratory variants in that spirit: not as a resolved solution, but as concrete directions for reducing the cost of sparsification on distributed semantic behavior while preserving feature readability.

4 Geometric Analysis of the Trade-off

The main results show that the trade-off is non-monotonic in k; this section asks what that non-monotonicity looks like geometrically. The key geometric question is where the best balance lies between functional preservation, sparse simplification, and avoidance of adverse geometry. We summarize the geometry along three axes:

1. **Functional preservation**, which captures whether the TopK LM still behaves like the dense LM at the output and block levels, using quantities such as logit cosine, top-1 agreement, top-5 overlap, and block CKA.
2. **Sparse simplification**, which captures whether the representation becomes more selective and is supported by fewer active coordinates, together with more concentrated MLP feature geometry, using quantities such as active-support fraction and MLP effective rank.
3. **Adverse-geometry avoidance**, which captures whether the TopK LM avoids the stronger geometric effects of the TopK constraint, such as the activation bulk shifting toward the negative side before TopK gating, residual bulk collapse, and highly imbalanced selected support, using quantities such as negative pre-TopK shift and sparse-support $p99$.

For visualization, each constituent metric is monotone-rescaled within the compared family and then aggregated within its axis; Appendix Section E.8 gives the exact metric definitions and aggregation rules. The adverse-geometry indicators also matter interpretability-side: once activation mass is pushed into harsher tails and more imbalanced selected supports, the surviving latent set becomes less clean, so brittle or overly generic coordinates are more likely to contaminate the concept-level feature bank that FADE evaluates. Under that reading, this section identifies the best balance point of the trade-off instead of mechanically favoring the smallest possible k.

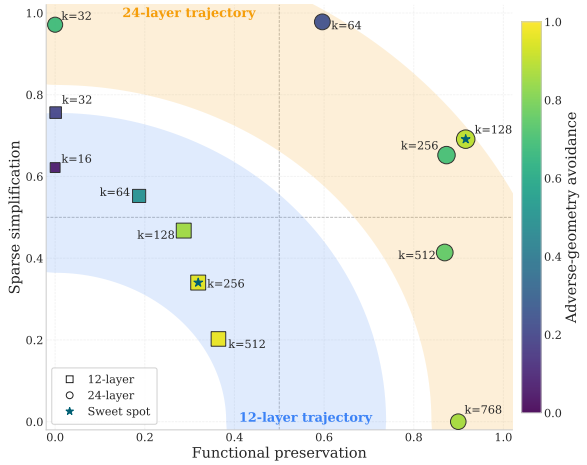
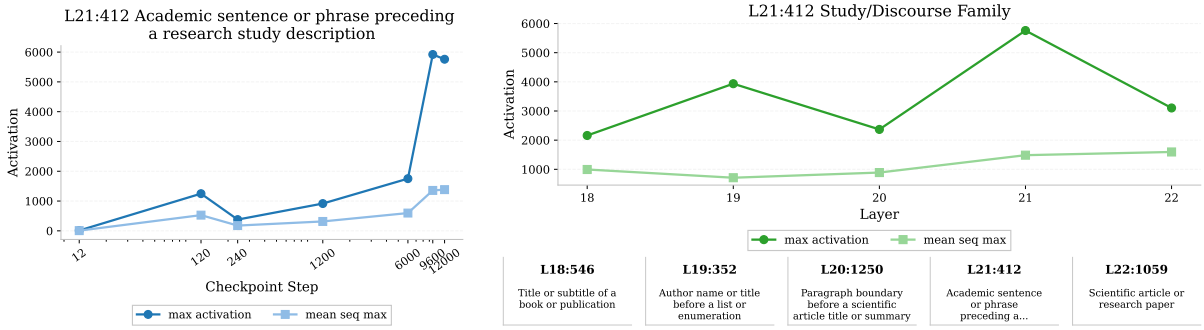


Figure 4: Geometric summary of the TopK trade-off. Each family is read along three axes: functional preservation, sparse simplification, and adverse-geometry avoidance. Decreasing k first sharpens sparse structure but eventually strengthens geometric distortions, so the favorable operating point is family-dependent rather than simply the smallest possible k.

Geometrically, lowering k first makes most activations smaller and more concentrated, but eventually creates stronger distortions: a smaller set of extreme activations becomes more prominent, and the selected coordinates become more unevenly used. The relevant operating point is therefore the most favorable trade-off point, where function remains strong, sparse structure is still substantial, and these distortions have not yet become too strong. Under that reading, lowering k initially moves the model toward a more human-readable sparse basis, but beyond the sweet spot the stronger pre-selection negative shift and support-tail amplification suggest a more brittle sparse regime in which reduced distributed semantic capacity degrades both task performance and feature quality. Figure 4 summarizes these three axes and marks the resulting balance point for each family.

The 24-layer family gives the clearest example.



(a) Checkpoint-wise exact tracing for $L21{:}412$

(b) Layer-wise semantic alignment around $L21{:}412$

Prompt	Baseline completion	Steered completion
I think	that a great way to get around that is to change your name. If you're a name and a person name, you can't just go out and write a new name.	that a great way to get **research into the science and research community** is to use **studies that have shown a lot of results** in a variety of ways. I've been looking at **studies** that...

(c) Direct steering with $L21{:}412$ on a neutral prompt

Figure 5: **Qualitative evidence for the article / study / publication family in the 24-layer $k = 128$ TopK LM.** Panels (a) and (b) use $L21{:}412$ as the main tracing case, showing that this latent becomes clearly readable only after substantial training and then differentiates across layers into related discourse roles. Here, *max activation* is the single largest token-level activation over the evaluation set, while *mean seq max* averages the within-sequence maxima and therefore reflects a more typical per-sample strength. Panel (c) shows that the same latent also directly steers a neutral prompt from non-academic text into study-oriented discourse.

Across $k \in \{64, 128, 256, 512, 768\}$, the TopK LMs remain close to the dense LM reference, so the sweep preserves substantial function throughout. But the smaller-k end shows stronger distortions in residual, MLP, and support geometry. Within this family, $k = 128$ is the smallest setting that still retains near-best output- and block-level similarity while avoiding the stronger distortions that emerge at $k = 64$.

The 12-layer family follows the same logic with the balance point shifted upward. There, $k = 256$ preserves function more cleanly than $k = 128$ and smaller settings while still retaining substantial sparse simplification. Below $k = 256$, the added sparsity fails to offset the weaker function preservation and stronger distortions. Taken together, the 12-layer and 24-layer results support the same geometric picture. In both families, decreasing k first sharpens sparse structure but eventually introduces stronger geometric distortions, so the best point is the one where function preservation and sparse simplification remain well balanced. What differs across families is where that point lies: near $k = 256$ in the 12-layer model and near $k = 128$ in the 24-layer model. This suggests that the most favorable TopK regime is family-dependent. More broadly, the TopK LMs preserve a large fraction of the dense LM's function while reorganizing computation into a sparse coordinate regime. That is why we read the geometry as a reorganization of dense computation into a sparse coordinate system instead of simple collapse or generic instability. Appendix Section E.8 gives the detailed width, alignment, and checkpoint-robustness summaries supporting this interpretation.

5 Qualitative Analysis and Showcase

After the quantitative and geometric analyses, we turn to the strongest qualitative evidence from the 24-layer $k = 128$ TopK LM. The main question is what this favorable sparse regime buys concretely: readable features that can also be traced over training and steered at inference time. Figure 5 centers that story on one main latent, while the appendix collects the broader prompt sweeps and supplementary formation examples.

5.1 Readable Latent Examples

For the qualitative analysis, we focus on one especially clear family: article / study / publication discourse. We therefore center Figures 5a and 5b on $L21{:}412$, a latent in the article / study / publication family, described as an academic sentence or phrase preceding a research study description.

Nearby cases $L17{:}791$, $L11{:}1403$, and $L18{:}546$ reinforce the same family through adjacent discourse roles such as study description, article summary, and publication-title language.

This family also illustrates one of the main advantages of a built-in sparse basis. Across checkpoints, the same layer:latent can be followed exactly, with no need to train and align a separate post-hoc dictionary at each checkpoint. Across layers, semantically related latents can be compared directly within the model's explicit sparse basis. In Figures 5a and 5b, $L21{:}412$ becomes clearly readable after substantial training, while neighboring article and publication features differentiate across depth instead of remaining identical copies. The appendix gives the full prompt sweeps for the main-family cases and additional formation examples beyond this central tracing pair.

5.2 A Steering Case Study

We again use $L21{:}412$ as the main steering example. In Figure 5c, a neutral "I think" prompt shifts from generic advice into explicit research-discourse prose about studies and results. This is a useful case because the baseline is clearly non-academic, the intervention produces a recognizably study-like continuation, and the same latent is also the paper's clearest tracing case. The neighboring cases $L17{:}791$, $L11{:}1403$, and $L18{:}546$ show that this coordinate belongs to a broader article / study / publication family; the appendix collects fuller prompt-wise sweeps for those cases together with additional positive examples from other discourse, topical, and lexical families. We therefore read the qualitative evidence conservatively. The 24-layer $k = 128$ TopK LM contains readable, traceable, and directly steerable features, with the strongest evidence concentrated in a limited set of families rather than spread uniformly across all latents.

6 Discussion

TopK LMs make the trade-off between capability and interpretability unusually explicit. Because the sparse basis is built into the model itself, the same architecture can be evaluated for downstream behavior, geometric reorganization, and feature-level readability. Our main result is a non-monotonic regime in which sparse features become easier to read while broader capability eventually degrades.

The task results, geometric analyses, and qualitative case studies all support the same interpretation. The main bottleneck appears to come primarily from applying hard TopK too uniformly across all layers. On that reading, TopK LM is useful both as an interpretable architecture and as a setting for studying how sparse, human-readable features emerge and what they cost.

A natural next step is to move toward more adaptive sparse designs. The appendix points in this direction, but broader design-space work is still needed to test how much interpretability gain can be retained while reducing the capability cost of the current all-layer setup.

7 Related Work

In the context of neural networks, *sparsity* has several related but distinct meanings (Farina et al., 2024): sparse model weights (Louizos et al., 2018; Mocanu et al., 2018; Evci et al., 2020; Jayakumar et al., 2021; Hunter et al., 2021), sparse attention (Child et al., 2019; Gupta et al., 2021), sparse activation of components in Mixture-of-Experts-style architectures (Shazeer et al., 2017; Jaszczur et al., 2021), and SAE-style representational sparsity of hidden state activations (Makhzani and Frey, 2015; Ahmad and Scheinkman, 2019; Hunter et al., 2021; Sharkey et al., 2023; Cunningham et al., 2023; Templeton et al., 2024; Gao et al., 2024). Our work falls into the last category. Closest among related work on representational sparsity are codebook feature layers (Tamkin et al., 2023), which, similar to our TopK layers, also enjoy inherent interpretability, but differ in the way sparsity is achieved, since their method relies on vector quantization and maximum inner product search, while our proposed architecture requires only minimal modification of the transformer architecture.

8 Conclusion

We have presented TopK LMs as a sparse language-model architecture in which interpretability is built into the model instead of being added afterward through a separate interpretation method. Our main result is that TopK LMs expose a non-monotonic performance-interpretability trade-off: broader capability is strongest at larger k, while the clearest interpretability gains appear at intermediate sparsity. Quantitative evaluations, geometric analyses, and qualitative case studies support the same picture. TopK LMs are therefore a concrete architecture for measuring, tracing, and improving sparse features directly.

Limitations

We studied a relatively narrow family of sparse language-model designs, centered on all-layer TopK activation with a small number of placement and training variants. Our conclusions should therefore be read as claims about this design family rather than about sparse language models in general. In particular, layer-selective sparsification, hybrid dense-sparse placement, and softer sparse mechanisms may realize different capability-interpretability trade-offs than the ones reported here.

Our experiments are also limited in scale. The main models are around the 1B-parameter regime, and extending the study to substantially larger models would require not only training additional systems but also repeating sparsity and optimization search at each new scale. Even in the current setup, the 12-layer and 24-layer experiments, including hyperparameter tuning, consumed approximately 1,400 GPU-hours (about 175 node-hours on an $8\times$H200 cluster). It therefore remains unclear how the most favorable TopK settings and geometric distortions would change at substantially larger scales. In particular, larger models increase the representational width d_{model}, which could either mitigate some geometric distortions by providing more room in representation space or intensify them by inducing harsher competition under TopK gating. The present results therefore do not establish that the same sparse regime would remain optimal, or even viable, for much larger foundation models.

Finally, our evaluation scope is still limited. We focus on language modeling, standard reasoning-style benchmarks, geometric analyses, and targeted qualitative interpretability case studies, but we do not yet evaluate long-context behavior, multimodal settings, or a broader range of specialized downstream domains. These are important directions for testing how broadly the present interpretability findings carry beyond the current experimental setting.

Ethical Considerations

This work studies a sparse language-model architecture intended to make internal features easier to inspect, trace across checkpoints, and intervene on. The paper does not introduce human-subject experiments or new sensitive-data collection, and so it does not raise unusual privacy concerns beyond the standard ones already associated with language-model pretraining and evaluation.

At the same time, model-native sparse features create a dual-use concern. The same properties that make TopK LMs easier to analyze could also support more targeted control of model behavior in larger systems. We therefore present these methods as tools for interpretability, auditing, and scientific analysis, not as deployment recipes for behavior manipulation. More broadly, improved interpretability can support safer and more accountable model development, but it does not by itself guarantee safe downstream use.

We also reviewed the licenses and terms of use for the main third-party artifacts used in this work, including model components, the training and evaluation framework, and the benchmark datasets. To the best of our understanding, their use in this research is consistent with the corresponding terms. We do not redistribute third-party artifacts beyond what their licenses permit.

We also used AI assistants during code development and manuscript preparation. All generated code and text were reviewed, revised, and verified by the authors before inclusion in the final paper.

References

Subutai Ahmad and Luiz Scheinkman. 2019. How can we be so dense? the benefits of using highly sparse representations.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan SU, Markus Kliegl, ZIJIA CHEN, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. 2026. Nemotron-CLIMB: Clustering-based iterative data mixture bootstrapping for language model pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda

Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR.

Mirko Farina, Usman Ahmad, Ahmad Taha, Hussein Younes, Yusuf Mesbah, Xiao Yu, and Witold Pedrycz. 2024. Sparsity in transformers: A systematic literature review. *Neurocomputing*, 582:127468.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders.

Laura Graesser, Utku Evci, Erich Elsen, and Pablo Samuel Castro. 2022. The State of Sparse Training in Deep Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 7766–7792.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. 2021. Memory-efficient transformers via top-k attention.

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.

Kevin Lee Hunter, Lawrence Spracklen, and Subutai Ahmad. 2021. Two sparsities are better than one: Unlocking the performance benefits of sparse-sparse networks. *CoRR*, abs/2112.13896.

Tatsuro Inaba, Kentaro Inui, Yusuke Miyao, Yohei Oseki, Benjamin Heinzerling, and Yu Takagi. 2025. How llms learn: Tracing internal representations with sparse autoencoders.

Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Łukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. 2021. Sparse is enough in scaling transformers.

Siddhant M. Jayakumar, Razvan Pascanu, Jack W. Rae, Simon Osindero, and Erich Elsen. 2021. Top-kast: Top-k always sparse training. *arXiv preprint arXiv:2106.03517*.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. Muon: An optimizer for hidden layers in neural networks.

Andrej Karpathy. 2025. nanochat: The best chatgpt that $100 can buy.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*.

Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. 2020. Soft Threshold Weight Reparameterization for Learnable Sparsity. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 5544–5555.

Denis Kuznedelev, Eldar Kurtic, Eugenia Iofinova, Elias Frantar, Alexandra Peste, and Dan Alistarh. 2024. Accurate neural network pruning requires rethinking sparse optimization. *Transactions on Machine Learning Research*.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. 2025. Sparse autoencoders do not find canonical units of analysis.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Christos Louizos, Max Welling, and Diederik P Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*.

Alireza Makhzani and Brendan J Frey. 2015. Winner-take-all autoencoders. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, Antonio Liotta, and et al. 2018. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(2383).

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. 2025. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*.

Gonçalo Paulo and Nora Belrose. 2025. Sparse autoencoders trained on the same data learn different features.

Bruno Puri, Aakriti Jain, Elena Golimblevskaia, Patrick Kahardipraja, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. 2025. FADE: Why bad descriptions happen to good features. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17138–17160, Vienna, Austria. Association for Computational Linguistics.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. 2024. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Lee Sharkey, Dan Braun, and Beren Millidge. 2023. Taking features out of superposition with sparse autoencoders. Accessed: 2025-05-15.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1690–1712, Suzhou, China. Association for Computational Linguistics.

Alex Tamkin, Mohammad Taufeeque, and Noah D Goodman. 2023. Codebook features: Sparse and discrete interpretability for neural networks. *arXiv preprint arXiv:2310.17230*.

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, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*.

Yang Xu, Yi Wang, and Hao Wang. 2025. Tracking the feature dynamics in llm training: A mechanistic study.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

A Expanded Related Work

A.1 Sparse Representations for Interpretability

In the context of neural networks, *sparsity* has several related but distinct meanings (Farina et al., 2024). The line of work most directly connected to ours is representational sparsity: instead of making weights or routing decisions sparse, it seeks sparse hidden-state coordinates that can be interpreted as readable features. This perspective appears in earlier sparse coding and winner-take-all style models (Makhzani and Frey, 2015; Ahmad and Scheinkman, 2019; Hunter et al., 2021) and, in mechanistic interpretability, is now most strongly associated with sparse autoencoders (SAEs) trained on language-model activations (Sharkey et al., 2023; Cunningham et al., 2023; Templeton et al., 2024; Gao et al., 2024). Our work is closest in motivation to this literature, but differs in where the sparse basis lives: SAE approaches learn a post-hoc sparse dictionary for a dense model, whereas TopK LM builds sparsity directly into the model's hidden representation itself. Codebook feature layers are also closely related because they provide model-native interpretable features without a separate post-hoc interpretation method (Tamkin et al., 2023), but they rely on vector quantization and maximum inner product search, while our method is a much lighter architectural modification.

A.2 Sparse Neural Architectures

Sparse structure is also widely studied for reasons other than post-hoc interpretability. Prior work has explored sparse model weights (Louizos et al., 2018; Mocanu et al., 2018; Evci et al., 2020; Jayakumar et al., 2021; Hunter et al., 2021), sparse attention patterns (Child et al., 2019; Gupta et al., 2021), and sparse activation of components in Mixture-of-Experts-style architectures (Shazeer et al., 2017; Jaszczur et al., 2021). These lines of work show that sparsity can be computationally useful or architecturally scalable, but their primary objective is usually efficiency, capacity, or routing, not exposing a directly readable hidden basis. Our setting is therefore closer to *interpretability-oriented* representational sparsity than to sparsity-for-efficiency work, even though all of these methods manipulate which parts of the network are active on a given input.

A.3 Capability–Interpretability Trade-offs

Our paper is motivated by the view that sparsity is not a free interpretability gain. Post-hoc SAE work has shown that highly readable sparse features can often be extracted from dense models (Sharkey et al., 2023; Cunningham et al., 2023; Templeton et al., 2024; Gao et al., 2024), while the broader sparsity literature makes clear that imposing sparse structure can change what a network can represent or compute (Farina et al., 2024). TopK LM brings these themes together by making the sparse basis intrinsic to the language model itself, which turns the central question from "can a sparse interpretation method be trained?" into "what performance cost is paid when the model must speak in sparse coordinates natively?" This framing is why we emphasize a performance-interpretability trade-off instead of claiming that sparse hidden states are universally better: in our experiments, increasing the TopK budget improves capability, while the clearest feature readability appears at intermediate sparsity. On the evaluation side, this also fits recent movement toward automated feature-interpretability benchmarks and metrics rather than purely anecdotal showcase examples (Karvonen et al., 2025; Paulo et al., 2025; Wu et al., 2025; Puri et al., 2025).

A.4 Training Dynamics and Feature Formation

This model-native sparse basis also connects our work to research on training dynamics, feature formation, and curriculum-sensitive representation learning in language models. In dense models, studying how a concept emerges over training typically requires separate post-hoc analysis at each checkpoint, together with an additional alignment step across checkpoints or layers (Xu et al., 2025; Inaba et al., 2025). By contrast, TopK LM exposes a fixed sparse coordinate system that can be followed directly during optimization, making it easier to ask developmental questions such as which linguistic features emerge early, which sharpen only late in training, and how data ordering or curriculum might shape that emergence process. We therefore view TopK LM not only as an interpretable architecture, but also as a useful experimental setting for studying the formation of human-readable linguistic features during LM training.

B Additional Architectural Variants and Search Space

This appendix summarizes architectural variants and hyperparameter explorations that are not part of the main fixed-k comparison.

B.1 Additional Architectural Variants

Alternate comparison protocols We also explored alternate comparison setups, most notably IsoFLOP-matched training comparisons and variants with an additional post-TopK ReLU. These experiments were useful during model selection and helped establish that trade-off-like behavior was not unique to a single recipe. However, they do not provide as clean a control as the no-postact formulation, because once a post-TopK nonlinearity is retained, the $k = d$ case no longer coincides exactly with the dense baseline. For that reason, the main quantitative claims are anchored on the TopK-only setup in which sparser models are allowed longer training horizons, while the IsoFLOP and ReLU-based comparisons are treated here as auxiliary evidence.

Annealed TopK smoothing Aiming to improve training stability and convergence, we also experimented with varying the degree of sparsity throughout training. The motivation was to transition gradually from a dense model to a TopK-sparse model, rather than imposing hard sparsity from the first update. The annealed TopK function introduces an annealing factor $\alpha \in [0, 1]$ that decays over training:

$$y = \alpha\, f(x) + (1 - \alpha)\big(f(x) \odot \mathbf{1}_{\{\, x \geq \tau_k(x) \}}\big) \quad (3)$$

where $f(x) = (f(x_1), \ldots, f(x_d))$ and $\odot$ denotes element-wise product. When $\alpha = 1$, the block is fully dense, and when $\alpha = 0$, the activation reduces to standard TopK. This approach is related to dense-to-sparse training (Kusupati et al., 2020; Graesser et al., 2022), but is treated here as an exploratory training variant rather than part of the main claim.

Hybrid block placement We also explored models in which the final few transformer blocks remain dense while earlier blocks use TopK sparsity. Let n_{nontopk} denote the number of final blocks left dense. Given a total of L layers, the first $L - n_{\text{nontopk}}$ layers use TopK blocks and the final n_{nontopk} layers remain conventional transformer blocks. This variant is useful for testing whether leaving later layers dense changes the quality-interpretability balance, but it is not part of the main fixed-k comparison.

Linear expansion variants We also explored linear-expansion variants that apply sparsity in an expanded latent space before projecting back to the model width. These variants can improve quality in some settings, but they do so by introducing additional parameters and architectural complexity beyond the plain TopK design. Because the goal of the main comparison is to isolate the trade-off induced by TopK sparsity itself, we do not include linear expansion in the main comparison.

B.2 Search Dimensions

In this section, we summarize the main hyperparameter and architecture-search axes that were explored around the core TopK design. The search dimensions that informed the final configuration include:

- sparsity level, including fixed-k sweeps over the TopK budget
- annealing schedules, including dense-to-sparse interpolation during training
- the number of final dense blocks, i.e., last non-TopK blocks
- TopK block placement across depth, including early, middle, late, and full-layer placement choices
- layer-wise k schedules, including variants that use different TopK budgets in early, middle, and late layers
- k-curriculum variants, in which the TopK budget changes during training
- post-activation choice after thresholding
- residual add-back and related sparse-block design choices
- hidden width and related width-allocation settings

Across these explorations, the configuration used in the main body was selected because it yielded the clearest fixed-k trade-off under a common training budget and a common interpretability protocol. In particular, the manuscript emphasizes the fixed-k, full-layer comparison with TopK-only gating and no residual add-back, while treating the other settings as exploratory side paths rather than central claims.

C Training Recipe Details

This appendix collects low-level training constants that are useful for reproducibility but not necessary for reading the main comparison.

Tokenizer and Pre-training Data All compared models use the same tokenizer learned within the nanochat preprocessing pipeline and then held fixed across dense and TopK variants. For pre-training data, we use Nemotron-ClimbMix (Diao et al., 2026), a compact high-quality corpus constructed with the CLIMB data-mixture algorithm from large web-scale sources.

Model Recipe Constants In the current canonical recipes, we use an attention head width of 128, a fixed nanochat width configuration corresponding to an internal aspect-ratio setting of 64, a maximum sequence length of 512 tokens, and a target parameter-to-data ratio of 8.

Training Recipe Constants For the main trade-off comparisons, the global batch size is $1{,}048{,}576 = 2^{20}$ tokens per optimization step. Under the current 8-GPU setup, the per-device batch size is 32. The WSD schedule uses a short warmup phase, a stable middle phase, and a warmdown phase over the final 20% of training. Other low-level launch and data-layout details follow the nanochat recipes associated with the experiments described in the main text.

Optimization Details All dense and TopK models are trained from the same nanochat entrypoint so that optimizer, scheduler, and checkpoint behavior remain aligned across families. Optimization uses Muon (Jordan et al., 2024) and AdamW (Loshchilov and Hutter, 2019).

Implementation Details The experiments are implemented on top of nanochat (Karpathy, 2025), with additional integration of the TopK, checkpointing, SAE-training, and FADE-evaluation components used in these experiments. All experiments are conducted on a single node with eight NVIDIA H200 SXM GPUs, each with 141 GB of VRAM. PyTorch Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) is used to shard model parameters, gradients, and optimizer states across devices, with no tensor parallelism. Forward and backward computations are executed in bfloat16 mixed precision.

C.1 Downstream Task Evaluation Details

Table 3 summarizes the representative task subset, including the intended task family, scoring type, few-shot counts, and the continuation delimiter used for answer scoring.

All task datasets are first shuffled with seed 1337 before optional subsampling. For the reported 24-layer results, training-time evaluation uses a fixed 500 examples per task snapshot rather than a dedicated full-dataset rerun, and the same 500-example setting is used across all compared models. Few-shot exemplars are sampled from the same task dataset with deterministic seed $1234 + \texttt{idx}$, excluding the current example itself from the candidate pool.

Scoring depends on the task type. For `multiple_choice` tasks such as ARC-e, PIQA, and HellaSwag, we score each answer option by its mean autoregressive loss over the answer span and choose the option with the lowest mean loss. For `language_modeling` tasks such as BBH CS Algorithms and CoQA, we score by exact token match on the continuation span, counting an example as correct only when the full continuation matches. The paper tables report raw accuracies for the five representative tasks; the broader CORE aggregate used elsewhere in training logs is the mean of centered task accuracies after adjusting each task by its official random baseline.

D Interpretability Pipeline Details

D.1 Baseline SAE Training Details

Dense baselines do not expose sparse latents directly, so we train a separate SAE for each compared baseline layer. All SAE and FADE analyses use the same base pretraining corpus as model training, namely the nanochat dataset derived from ClimbMix-400B. Within that shared corpus, SAE training uses the training split, while SAE diagnostics, latent-record extraction, and FADE evaluation all use held-out text from the validation split. These SAEs are trained with the native nanochat trainer and then reused as fixed anchors for record extraction and FADE evaluation. In the experiments reported in the main text, the baseline SAE setup uses a TopK SAE with $k = 32$, latent width $32\times$ the hidden width, learning rate 10^{-3}, 1000 warmup steps, 100M training tokens, and 10M held-out evaluation tokens. For the dense anchors used in the main family comparisons, this means training one full SAE bank for the 12-layer dense endpoint

Table 3: Representative downstream-task protocol used for the reported d24 tables. These are CORE-based representative task accuracies computed through the native nanochat evaluation pipeline.

Paper label	Task family	Scoring type	Few-shot	Random baseline	Delimiter
ARC-e	world knowledge	multiple_choice	10	25.0	`\nAnswer:`
PIQA	commonsense reasoning	multiple_choice	10	50.0	`\nAnswer:`
HellaSwag	language understanding	multiple_choice	10	25.0	
BBH CS Algo	symbolic problem solving	language_modeling	10	0.0	
CoQA	reading comprehension	language_modeling	0	0.0	

and another for the 24-layer dense endpoint, rather than evaluating Dense+SAE through a lightweight or intentionally weak control. The encoder is linear, the decoder is linear with a learned bias term, the encoder nonlinearity is ReLU, and the decoder columns are renormalized to unit norm during training. The training-side defaults also use a refresh batch size of 32, an SAE batch size of 32768, a 3M-vector activation buffer, `auxk_alpha` $= 1/32$, and a dead-feature threshold of 10M tokens. The resulting dense-anchor setup is summarized in Table 4.

Activation preprocessing is also fixed across layers. We remove BOS tokens from packed sequences, flatten token activations to a two-dimensional $[N, d]$ matrix, and normalize hidden states before SAE training and evaluation. The trainer further stabilizes learning with geometric-median initialization, `auxk_loss`, long-horizon dead-feature tracking, decoder-gradient projection, and gradient clipping. These details matter because the dense baseline's sparse features are produced by the SAE rather than by the language model itself, so we want to avoid handicapping the dense anchor with an obviously undertrained SAE.

Aggregate diagnostics for both dense-anchor banks are reported in Table 5. The 12-layer dense bank is uniformly healthy by reconstruction-side diagnostics: held-out cosine similarity stays in the range $0.9617 - 0.9874$, R^2 in the range $0.9247 - 0.9748$, and fraction alive in the range $0.9967 - 1.0000$. For that family, the dense+SAE anchor is therefore a strong and fair comparison baseline rather than a weak post-hoc control. The 24-layer dense bank is weaker, especially in early-to-middle layers, but it is still not a trivial or visibly broken control: held-out cosine similarity remains $0.9449 - 0.9871$, R^2 remains $0.8927 - 0.9744$, and relative reconstruction bias stays essentially neutral at $1.0001 - 1.0011$. Its main limitation is feature utilization rather than a gross reconstruction-scale failure, with fraction alive ranging from 0.7571 to 0.9267 and dead features concentrated in the lower and middle layers. In both banks, the strongest held-out reconstruction quality appears near the top layers. We therefore treat the dense+SAE anchors as serious, intentionally trained baselines; the 12-layer anchor is strong throughout, while the 24-layer anchor remains usable but should be read with this reconstruction-quality caveat in mind.

D.2 FADE Evaluation Details

The FADE pipeline has three stages. First, we extract activation records for each target layer. Second, we generate one short natural-language description per latent from those records. Third, FADE uses that description to generate synthetic sequences and to rate how well the latent's activation behavior matches the described concept. In our implementation, this pipeline is executed through `fade-language` v0.1.2 with a local nanochat adapter and small robustness patches around missing-score handling.

Record extraction and FADE natural-text documents both come from the validation split. For the main comparisons, we use independent FADE with 100 target latents and 64 recording batches, with latent ranking based on activation frequency. The extracted records store high-activation snippets around activation peaks, and the resulting latent bank is then passed to the description and FADE stages without any additional retraining. Description generation uses deterministic settings, while FADE generation and rating use separate decoding settings so that concept generation remains diverse but rating remains stable. For the reported experiments, the description-generation backend is a locally served Meta Llama-3.1-8B-Instruct model.

Among recent automated feature-interpretability evaluations, we use FADE as the concrete protocol in this paper (Puri et al., 2025). FADE returns four per-feature scores: clarity, responsiveness, purity, and faithfulness. In intuitive terms, clarity asks whether a feature separates natural examples

Table 4: Training setup for the dense-anchor SAE banks used in the dense+SAE versus TopK comparisons. Each dense endpoint uses a separate SAE for every layer, all trained with the same recipe rather than a weaker or ad hoc post-hoc control. Both banks use SAE $k = 32$, 100M train tokens, 10M held-out tokens, learning rate 10^{-3}, and 1000 warmup steps.

Family	Layers	Hidden dim	Latents	Width
12-layer $k = 768$	12	768	24,576	32×
24-layer $k = 1536$	24	1536	49,152	32×

Table 5: Aggregate reconstruction-side diagnostics for the dense-anchor SAE banks. Held-out values and train-side health are reported as min / mean / max across layers. Relative reconstruction bias staying near 1.0 indicates that the main difference between the two banks is reconstruction quality and feature utilization rather than a gross scale mismatch.

(a) Held-out reconstruction diagnostics

Family	Eval loss	Cosine	R^2	Var. expl.	Alive	Dead feat.
12-layer	0.025 / 0.054 / 0.075	0.962 / 0.973 / 0.987	0.925 / 0.946 / 0.975	0.821 / 0.861 / 0.950	0.997 / 0.999 / 1.000	0 / 25.3 / 82
24-layer	0.026 / 0.076 / 0.107	0.945 / 0.961 / 0.987	0.893 / 0.924 / 0.974	0.780 / 0.806 / 0.934	0.757 / 0.828 / 0.927	3604 / 8469 / 11939

(b) Train-side health

Family	Dead last	Alive last	Rel. recon. bias
12-layer	0.00 / 0.56 / 2.47	0.999899 / 0.999977 / 1.000000	1.000176 / 1.000385 / 1.000663
24-layer	151.29 / 573.87 / 1050.53	0.978627 / 0.988325 / 0.996922	1.000144 / 1.000737 / 1.001083

from synthetic examples generated from its description, responsiveness asks whether higher-rated concept examples activate the feature more strongly than non-concept examples, purity measures how cleanly high activations concentrate on the concept rather than on generic continuation patterns, and faithfulness measures whether direct intervention increases concept-related generation. Here, the main scalar is FADE's main summary score, computed as the unweighted mean of the valid clarity, responsiveness, and purity values. For the 12-layer fixed-k main comparison, this scalar is first averaged across selected latents within each evaluated layer and then averaged again across all 12 sparse layers. Representative early/middle/late layer summaries are retained only for supplementary layer-wise analysis and for the separate seed-robustness check. Because faithfulness is often missing under the smaller-budget evaluation protocol, it is treated as supplementary rather than part of the main ranking.

In the main FADE protocol, each evaluated layer contributes 100 target latents ranked by activation frequency. Activation records are collected over 64 batches using a recording device batch size of 4, and the FADE stage uses 64 synthetic documents together with 64 activation-side samples per feature. On the rating side, we use a subject batch size of 4, a rating batch size of 15, and a faithfulness-gate threshold of 0.5, although faithfulness itself is usually absent in the smaller-budget evaluations reported in the main text. Semantically matched FADE is evaluated separately and reported as the paper's fairness-oriented robustness readout. We use FADE in two modes. Independent FADE is the standard model-independent use of FADE, in which each model is evaluated on its own target latent set with no cross-model matching. Semantically matched FADE keeps the same FADE scoring pipeline but changes the evaluated latent set: for each comparison between a dense baseline and a TopK LM, we restrict attention to features whose descriptions align between the two models. Operationally, each compared layer first defines a larger frequency-ranked latent candidate pool, and we then form candidate pairs between dense and TopK features using a description/example-based heuristic matching score rather than a separate embedding model. In these experiments, this candidate pool contains the top 1000 latents per layer by activation frequency. For each comparison, we consider up to 200 matching candidates and retain up to 10 high-scoring matches per right-side latent, requiring a minimum match score of 0.15 before a pair is accepted. The final matched subset keeps only the strong-match band and is capped at 10 retained pairs per comparison layer, after which FADE is evaluated on that subset for both the dense

model and the corresponding TopK model. In the main text, this matched readout is averaged over all compared layers, while smaller representative-layer subsets are used only for supplementary comparisons. The matching score itself combines three terms: a heavily weighted overlap ratio on concept tokens, a Jaccard score on the same concept-token sets, and a lower-weight cosine similarity computed from token-count vectors extracted from the top activation examples. The design goal of this score is to build a concept-aligned subset rather than an index-aligned one. We therefore weight direct concept-token overlap most heavily, use the Jaccard term to avoid over-crediting very short descriptions, and keep the example-based cosine as a weaker auxiliary signal so that noisy example overlap does not dominate the match. This makes the matched subset a conservative semantic-alignment heuristic rather than an exact equivalence test. As a result, semantically matched FADE should be read as "FADE on a comparison-specific semantically aligned subset," not as a modified FADE score. The matched subset is comparison-specific rather than globally shared across all k, so it is useful for testing aligned feature retention relative to the dense baseline rather than for defining the paper's global independent FADE ranking. In the seed-robustness study, FADE-side randomness is held fixed so that the measured spread primarily reflects variation from training seed rather than evaluation-side stochasticity. For the 12-layer fixed-k series, Table 6 reports the corresponding layerwise independent FADE values across all 12 sparse layers.

E Additional Results

This appendix collects supplementary analyses that are useful for interpretation but are not part of the present mainline comparison.

E.1 Layerwise independent FADE Under the Mainline 12-Layer Protocol

For the present 12-layer mainline, Table 6 breaks the all-layer independent FADE summary into its per-layer values. This table uses the same main comparison setup as the main text and is included only to show how the average score is distributed across depth.

E.2 All-Layer semantically matched FADE Summary for the Mainline Sweeps

Table 7 reports the exact all-layer semantically matched FADE values summarized visually in Figure 3. We keep the table in the appendix because the main text is better served by the trend-level comparison, whereas the exact per-k values are mainly useful for detailed inspection and citation.

E.3 Seed Variation and Taskwise Spread in the Mainline Sweeps

For the representative five-task evaluations in the mainline sweeps, we also computed seed-robust summaries over evaluation seeds $42, 43, 44$. Table 8 reports the resulting mean and SD of the representative task average for the completed 12-layer and 24-layer series. These summaries should be read as evaluation-set and few-shot sensitivity rather than as training-seed variation, because the model weights are held fixed.

To complement those seed summaries, Table 9 reports taskwise spread across the completed k sweeps using seed-mean task scores. This makes it easier to separate two different sources of variation: cross-seed instability of the representative aggregate and systematic cross-k sensitivity of individual tasks. In the 12-layer sweep, ARC-e and CoQA are the most k-sensitive tasks, whereas BBH-CS is comparatively stable. In the 24-layer sweep, HellaSwag and CoQA show the largest cross-k spread, again with BBH-CS remaining the most stable task family.

E.4 12-Layer Results Under an IsoFLOP-Matched Comparison

Before the present 12-layer mainline was fixed to the longer-horizon sparse comparison used in the main text, we used an IsoFLOP-matched comparison with a shared 4k training budget across the fixed-k series. Table 10 reports the corresponding paper-style summary table. The qualitative picture already resembled the later mainline: the dense baseline was strongest overall, higher-k TopK models recovered most of the task-level quality, and interpretability peaked at an intermediate sparse point rather than at the highest-performance end. In particular, $k = 64$ remained the clearest trade-off point on the interpretability side, while $k = 512$ and $k = 768$ were the strongest performance-side TopK variants.

E.5 Seed Robustness Under the IsoFLOP-Matched Comparison

Before the current 12-layer mainline was fixed to the longer-horizon sparse comparison used in the

Table 6: Layerwise independent FADE values for the 12-layer fixed-k series in the main comparison. Values are reported for all sparse layers $L01$ through $L12$ under the same 12-layer main comparison setup as Table 1. The final row is an auxiliary post-hoc SAE readout on the $k = 768$ model and is not part of the main fixed-k ranking.

Model	Avg.	L01	L02	L03	L04	L05	L06	L07	L08	L09	L10	L11	L12
$k = 768$ (100%, baseline)	0.2647	0.3274	0.1616	0.1521	0.1886	0.2181	0.2464	0.2699	0.2900	0.3190	0.3363	0.3389	0.3274
$k = 768$ + SAE	0.3659	0.4001	0.4000	0.3833	0.4036	0.3753	0.3668	0.3516	0.3566	0.3507	0.3066	0.3453	0.3505
$k = 512$ (66.7%)	0.3993	0.4507	0.3856	0.2836	0.3590	0.3786	0.3931	0.4342	0.3846	0.4335	0.4173	0.4365	0.4343
$k = 256$ (33.3%)	0.4298	0.4539	0.4771	0.3965	0.4058	0.4223	0.4316	0.4194	0.4238	0.4220	0.4277	0.4332	0.4446
$k = 128$ (16.7%)	0.4557	0.4786	0.4339	0.4562	0.4426	0.4640	0.4548	0.4716	0.4749	0.4707	0.4396	0.4338	0.4476
$k = 64$ (8.3%)	0.4702	0.4676	0.4604	0.4796	0.4739	0.4701	0.4722	0.4886	0.4676	0.4518	0.4585	0.4448	0.5077
$k = 32$ (4.2%)	0.4585	0.4738	0.4358	0.4518	0.4957	0.4722	0.4679	0.4740	0.4482	0.4509	0.4395	0.4435	0.4488
$k = 16$ (2.1%)	0.4556	0.4534	0.4676	0.4643	0.4949	0.4930	0.4360	0.4397	0.4591	0.4797	0.4603	0.4777	0.3417

Table 7: All-layer semantically matched FADE summary for the mainline 12-layer and 24-layer comparisons. For each dense–TopK pair, the matched subset is recomputed from strong description-level matches, so the baseline-side matched score is comparison-specific. Positive deltas indicate that the TopK side scores higher than the corresponding SAE anchor on the semantically aligned subset.

(a) 12-layer anchor: $k = 768$+SAE

Model	Baseline	TopK	Delta
$k = 16$	0.4222	0.3619	-0.0603
$k = 32$	0.4217	0.4130	-0.0086
$k = 64$	0.4237	**0.4750**	**+0.0513**
$k = 128$	0.4312	0.4602	+0.0290
$k = 256$	0.4335	0.4355	+0.0021
$k = 512$	0.4310	0.4145	-0.0165
$k = 768$	0.4166	0.2738	-0.1428

(b) 24-layer anchor: $k = 1536$+SAE

Model	Baseline	TopK	Delta
$k = 32$	0.1839	0.4412	+0.2573
$k = 64$	0.1947	0.4475	+0.2527
$k = 128$	0.1926	**0.4608**	**+0.2682**
$k = 256$	0.1962	0.4563	+0.2601
$k = 512$	0.2079	0.4551	+0.2472
$k = 768$	0.1930	0.4329	+0.2399
$k = 1536$	0.2024	0.1817	-0.0207

Table 8: Seed-robust summary of the representative five-task average for the mainline 12-layer and 24-layer sweeps. For each completed k, the reported mean and SD are computed over evaluation seeds $42, 43, 44$.

12-layer sweep			24-layer sweep		
Model	Avg.	SD	Model	Avg.	SD
$k = 768$	44.03	0.48	$k = 1536$	47.15	0.28
$k = 512$	43.49	0.66	$k = 768$	47.41	0.26
$k = 256$	44.17	0.65	$k = 512$	47.16	0.51
$k = 128$	42.87	0.46	$k = 256$	46.28	0.34
$k = 64$	42.04	0.62	$k = 128$	49.52	0.40
$k = 32$	40.36	0.49	$k = 64$	47.83	0.75
$k = 16$	38.31	0.06	$k = 32$	45.85	0.51

Table 9: Taskwise spread across the completed mainline k sweeps. For each task, scores are first averaged over evaluation seeds $42, 43, 44$, after which 'SD' is the population standard deviation across the completed k values in that family and 'Range' is max minus min across the same sweep.

	12-layer sweep		24-layer sweep	
Task	SD	Range	SD	Range
ARC-e	3.49	11.53	1.86	5.73
PIQA	1.81	5.87	0.92	3.13
HellaSwag	1.61	4.60	1.45	4.80
BBH-CS	1.99	6.33	2.00	6.87
CoQA	2.40	6.80	1.95	6.27
5-task avg	2.01	6.11	1.10	3.67

main text, we ran a seed-robustness study under the IsoFLOP-matched comparison setup. That comparison reran three representative points with seeds $42/43/44$: the dense baseline, the interpretability-oriented TopK point $k = 64$, and the higher-performance TopK point $k = 512$. All compared models were evaluated with the same independent FADE setup, while FADE-side randomness was held fixed so that the observed spread primarily reflects training-seed variation.

The qualitative ordering remained stable across seeds. In performance, $k = 512$ consistently outperformed $k = 64$ on both validation loss (2.9319 ± 0.0032 vs. 3.1878 ± 0.0204) and representative downstream-task average (43.6 ± 0.5 vs. 39.4 ± 0.8). In interpretability, $k = 64$ consistently exceeded $k = 512$ in mean FADE (0.4729 ± 0.0126 vs. 0.4298 ± 0.0073). The dense baseline remained best in performance (2.8605 ± 0.0009) while achieving FADE comparable to $k = 64$ (0.4740 ± 0.0065 vs. 0.4729 ± 0.0126). As a sup-

Table 10: 12-layer IsoFLOP-matched results. FADE is the latest target-only mean over all 12 sparse layers; ‘k=768‘ uses native dense features, and ‘k=768 + SAE‘ is the width-matched random-768 SAE control.

	Language Modeling	Downstream Tasks						Interpretability
Model	Val. Loss ↓	ARC-e ↑	PIQA ↑	HellaSwag ↑	BBH-CS ↑	CoQA ↑	Avg. ↑	FADE ↑
$k = 768$	**2.862**	**58.00**	68.80	**37.40**	42.40	14.60	**44.24**	0.4079
$k = 768$ + SAE	–	–	–	–	–	–	–	0.3729
$k = 512$ (66.7%)	2.936	56.60	68.60	35.40	43.20	**16.60**	44.08	0.4230
$k = 256$ (33.3%)	3.004	54.20	66.80	32.20	**45.40**	12.20	42.16	0.4288
$k = 128$ (16.7%)	3.095	54.40	66.80	32.80	45.00	10.00	41.80	0.4513
$k = 64$ (8.3%)	3.204	46.60	64.60	29.80	42.60	9.00	38.52	0.4735
$k = 32$ (4.2%)	3.301	43.20	65.00	30.60	37.60	8.00	36.88	0.4646
$k = 16$ (2.1%)	3.502	38.60	59.80	29.40	44.00	4.00	35.16	**0.4754**

plementary matched-FADE check on the same representative points, TopK also remained favorable on average across seeds: for $k = 64$, matched TopK FADE exceeded the matched baseline value by 0.0359 ± 0.0322, and for $k = 512$, the corresponding advantage was 0.0380 ± 0.0332. Because those matched subsets are recomputed per model and per seed, we do not use them as the main ranking criterion, but they still provide supportive evidence that the IsoFLOP-matched trade-off pattern was not an artifact of a single seed.

E.6 Matched FADE Under the IsoFLOP-Matched Comparison

semantically matched FADE was also evaluated under the IsoFLOP-matched 12-layer comparison as a fairness-oriented complement to the main independent FADE readout. Under standard independent FADE, each model is evaluated on its own target latent set. That captures how many readable features the model exposes overall, but it can confound cross-model comparison because the compared latent sets need not represent the same concepts. In particular, a model that more readily forms neurons for surface-level or otherwise easily describable patterns can look better under a purely model-independent readout even when the compared semantic sets differ. semantically matched FADE addresses this concern by restricting each dense–TopK comparison to a semantically aligned subset built from description-level matches. We therefore use it as a fairness-oriented complement rather than as the main ranking, because the matched subset is recomputed separately for each comparison pair and no longer measures total feature yield over the whole model.

Under this matched protocol, the TopK models remained broadly competitive with the dense baseline and often compared favorably on their strongest aligned concepts. For the Pareto-front points $k \in \{64, 128, 256, 512, 768\}$, TopK exceeded the baseline at $k = 64$ (0.4718 vs. 0.4255), $k = 256$ (0.4873 vs. 0.4485), and $k = 768$ (0.4777 vs. 0.4128). The baseline was only slightly higher at $k = 128$ (0.4477 vs. 0.4384), and $k = 512$ was nearly tied (0.4211 vs. 0.4198). We therefore read this semantically matched FADE result as supportive evidence that high-performing TopK models can still retain a substantial subset of strongly aligned concepts under a more semantically controlled comparison, even though the current mainline comparison will need its own matched-FADE evaluation.

E.7 24-Layer Results Under an IsoFLOP-Matched Comparison

Before the present 24-layer scale-up check was moved to the longer-horizon sparse comparison used in the main text, we also ran a smaller IsoFLOP-matched comparison at the deeper scale. Table 13 reports the corresponding appendix-side summary table. The qualitative picture already pointed in the same direction as the later 24-layer mainline results: the dense baseline remained strongest overall, the higher-k sparse model $k = 512$ recovered much of the broad task-level quality, and the lower-k model $k = 64$ was more favorable on the trade-off side than on pure capability. At the same time, this deeper comparison was much narrower than the present fixed-k sweep, containing only the dense baseline and two representative sparse points. We therefore keep it as supplementary context rather than as part of the paper’s primary 24-layer evidence.

Table 11: Three-seed robustness summary for the representative 12-layer comparison points. This table reports the main metrics used in the seed-robustness analysis: validation loss, downstream-task average, and target-only FADE.

	Performance		Interpretability
Model	Val. Loss	Task Avg.	Target-only FADE
Baseline	2.8605 ± 0.0009	45.9 ± 1.2	0.4740 ± 0.0065
$k = 64$ (8.3%)	3.1878 ± 0.0204	39.4 ± 0.8	0.4729 ± 0.0126
$k = 512$ (66.7%)	2.9319 ± 0.0032	43.6 ± 0.5	0.4298 ± 0.0073

Table 12: Matched-subset seed-robustness summary for the representative 12-layer TopK points. Each row defines its own matched subset, and we report the dense-model score, the TopK score, and their difference on that same subset.

Model	Matched Dense FADE	Matched TopK FADE	TopK − Dense
$k = 64$ (8.3%)	0.4288 ± 0.0237	0.4646 ± 0.0130	0.0359 ± 0.0322
$k = 512$ (66.7%)	0.4141 ± 0.0181	0.4521 ± 0.0387	0.0380 ± 0.0332

E.8 Activation Geometry and Logit Alignment at the Deeper Scale

This appendix collects the 24-layer geometry summaries behind Section 4. The main reference is the matured dense $k = 1536$ model at 20k steps, and the comparison family is the completed sparse sweep $k \in \{64, 128, 256, 512, 768\}$. The main result is still asymmetric. The sparse models remain functionally close to the dense reference model, but their activation geometry is clearly reorganized. Section 4 refers to three geometric desiderata. In this appendix, we operationalize them as follows: functional preservation is read from logit cosine, top-1 agreement, and block CKA; sparse simplification is read from smaller active support and more concentrated MLP geometry, using statistics such as active-support fraction and MLP effective rank; and adverse geometry is read from harsher pre-selection shifts and extreme support tails, using statistics such as |pre-TopK median|, the negative part of pre-TopK $p90$, sparse-support $p99$, and local ID gaps.

- **Logit cosine**: token-weighted cosine similarity between dense and TopK full output-logit vectors on the same token positions.
- **Top-1 agreement**: fraction of evaluated token positions where the dense and TopK models predict the same top token.
- **Top-5 overlap**: overlap between the dense and TopK top-5 predicted token sets on the same token positions.
- **Block CKA**: similarity between corresponding dense and TopK block representations.
- **Active-support fraction**: fraction of TopK coordinates that remain nonzero after gating.
- **MLP effective rank**: concentration of the MLP representation across dimensions, computed from sampled hidden-state rows.
- **|Pre-TopK median|** and **negative pre-TopK** $p90$: shifts in the pre-TopK activation distribution, including whether the activation bulk moves toward the negative side.
- **Sparse-support** $p99$ and **local ID gaps**: tail-heaviness and local geometric imbalance in the selected support.

For a metric m defined over a comparison family $\mathcal{F}$, let $m_{\min} = \min_{k' \in \mathcal{F}} m(k')$ and $m_{\max} = \max_{k' \in \mathcal{F}} m(k')$. We first rescale to $[0, 1]$ by

$$\hat{m}(k) = \frac{m(k) - m_{\min}}{m_{\max} - m_{\min}},$$

and use $1 - \hat{m}(k)$ instead whenever smaller values are better. Write $c =$ logit cosine, $a =$ top-1 agreement, $b =$ block CKA, $s =$ active-support fraction, $r =$ MLP effective rank, $u =$ |pre-TopK median|, $v =$ negative pre-TopK $p90$, $w =$ sparse-support $p99$, and $g =$ local ID gap. The axis scores are then

$$\begin{aligned}
F(k) &= (\hat{c}(k) + \hat{a}(k) + \hat{b}(k))/3, \\
S(k) &= (\hat{s}(k) + \hat{r}(k))/2, \\
P(k) &= (\hat{u}(k) + \hat{v}(k) + \hat{w}(k) + \hat{g}(k))/4, \\
A(k) &= 1 - P(k), \\
T(k) &= F(k) + S(k) + A(k) - 1.
\end{aligned}$$

Here F, S, P, A, T denote functional preservation, sparse simplification, pathology penalty, adverse-geometry avoidance, and the final trade-off score. Table 14 summarizes the experimental setup used

Table 13: Results for a 24-layer IsoFLOP-matched comparison used in an earlier appendix-side scale-up check. All models share the same training horizon and representative-task protocol; the table reports broad performance together with complementary target-only FADE.

	Language Modeling	Downstream Tasks						Interpretability
Model	Val. Loss ↓	ARC-e ↑	PIQA ↑	HellaSwag ↑	BBH-CS ↑	CoQA ↑	Avg. ↑	FADE ↑
Baseline (+ SAE)	**2.443**	**72.20**	**76.20**	**57.40**	38.40	**28.60**	**54.56**	**0.4866**
$k = 64$ (4.2%)	2.827	61.60	71.00	43.20	42.60	15.00	46.68	0.4727
$k = 512$ (33.3%)	2.590	69.60	75.40	50.80	**42.80**	23.00	52.32	0.4504

for this analysis. Table 15 gives the family-level trade-off sweep across sparse k in the 12-layer and 24-layer families, while Table 16 reports the 24-layer output- and block-alignment sweep against the dense reference.

All geometry statistics are computed on the same shared evaluation-token set, taken from held-out text in the nanochat ClimbMix validation split. Operationally, we evaluate the dense and TopK models on the same token stream and compare residual/output statistics position by position, rather than aggregating over separately sampled prompts. For the primary paired analyses, we use 8 shared batches of size 8 with sequence length 512, yielding 32768 token positions before validity filtering. The activation-distribution summaries use 262144 sampled values per activation family, whereas the hidden-geometry, covariance, and PCA analyses use 4096 sampled hidden-state rows per analyzed layer and capture family. Table 15 should be read as the family-level sweep summary of that analysis. The picture is clearly family-dependent: the 12-layer family reaches its best sparse balance near $k = 256$, whereas the 24-layer family reaches it near $k = 128$. The 24-layer family also occupies a much more favorable region overall, with stronger functional preservation and sparse simplification while still maintaining high adverse-geometry avoidance.

Table 16 shows the 24-layer family-level complement to that score summary. Across the full completed sweep, logit cosine stays above 0.928, top-1 agreement stays above 0.719, and top-5 overlap stays above 0.666. Block CKA also remains substantial across all compared k, peaking at 0.8160 for $k = 768$. So even though internal geometry is reorganized strongly, the 24-layer sparse family still preserves substantial output and block-level similarity to the dense reference model.

We therefore read this 24-layer geometry result as supportive evidence for the overall interpretation. The deeper TopK models do not behave like the dense baseline internally, but they also do not look obviously unstable or abnormal. Instead, they implement a structured sparse regime in which much of the dense model's function is preserved while representational mass is redistributed into a smaller, heavier-tailed active support.

E.9 Continued Training at the Deeper Scale

We also tested whether longer training changes the 24-layer picture within the sparse family by continuing the 24-layer $k = 64$ TopK model from 10k to 20k steps. Table 17 shows that continued training improves optimization-side metrics inside the sparse family: validation loss improves from 2.8272 to 2.7795, and CORE improves from 0.1640 to 0.1852. However, representative downstream-task average changes only marginally ($46.68 \rightarrow 46.76$), and matched interpretability does not improve uniformly across depth. Under the new horizon-matched 20k comparison, the dense baseline still remains clearly stronger in broad performance (2.3913 vs. 2.7795 in validation loss, 56.76 vs. 46.76 in representative task average), but the matched-FADE advantage for TopK stays positive in all three layer bands. The middle-layer advantage strengthens substantially ($+0.0721 \rightarrow +0.1285$), the early-layer advantage remains modest ($+0.0353 \rightarrow +0.0316$), and the final-layer advantage becomes much smaller ($+0.1083 \rightarrow +0.0177$) rather than turning negative. This suggests that, at the deeper scale, continued sparse training improves capability more smoothly than it improves feature quality, while redistributing the strongest positive matched-feature signal toward the middle of the network instead of strengthening it uniformly across depth. Because the matched subsets are still recomputed separately at 10k and 20k, we continue to treat this as an appendix-scale observation and read the TopK-minus-dense delta within each horizon as the most

Table 14: Protocol summary for the 24-layer geometry analysis. The main reference is the matured dense $k = 1536$ model at 20k steps, compared against the completed sparse TopK sweep. Representative width statistics use the strongest sparse contrast $k = 64$, while matched-step and cross-k summaries are used as robustness checks.

Item	Setting
Dense reference	24-layer dense $k = 1536$ model at 20k steps
Sparse family compared	24-layer all-layer TopK models with $k \in \{64, 128, 256, 512, 768\}$, evaluated at their completed longer-horizon sparse checkpoints
Representative width contrast	Dense $k = 1536$ at 20k versus TopK $k = 64$ at 24k, with an additional checkpoint-aligned 20k-vs-20k check
Shared evaluation data	Held-out text from the ClimbMix-derived validation split, with identical token batches for all dense-vs-TopK comparisons
Default paired batch budget	8 batches, batch size 8, sequence length 512, giving $8 \times 8 \times 512 = 32768$ token positions before validity filtering
Hidden-geometry sampling	4096 sampled hidden-state rows per analyzed layer and capture family
Activation-distribution summaries	262144 sampled activation values per activation family
Output-alignment summaries	Token-weighted comparisons on the same token positions, including full-logit cosine, top-1 agreement, top-5 overlap, block CKA, and block aligned cosine
Global geometry summaries	Effective-rank and PCA-based summaries from 4096 sampled rows per layer, averaged across layers for overall statistics
Checkpoint robustness	Matched-step dense-vs-TopK comparisons at 5k, 10k, 15k, and 20k to separate early sparse-bulk effects from later tail amplification
Aggregation	Alignment metrics are token-weighted over evaluated positions; layerwise geometry summaries are averaged arithmetically when an overall scalar is needed

Table 15: Family-level trade-off summary across sparse k for the 12-layer and 24-layer families. Each score is computed relative to the dense endpoint within that family ($k = 768$ for the 12-layer family and $k = 1536$ for the 24-layer family). sparser models are trained longer under the family-specific schedule described in the main setup. Adverse-geometry avoidance is $1-$pathology penalty.

Family	k	Functional	Sparse simp.	Adv.-geom. avoid.	Tradeoff
12-layer	16	0.000	0.622	0.056	-0.322
	32	0.001	0.756	0.197	-0.046
	64	0.187	0.552	0.496	0.236
	128	0.287	0.468	0.853	0.608
	256	0.319	0.340	0.957	0.616
	512	0.364	0.202	0.968	0.535
24-layer	32	0.000	0.971	0.675	0.646
	64	0.596	0.978	0.234	0.808
	128	0.916	0.691	0.892	1.499
	256	0.873	0.652	0.694	1.219
	512	0.869	0.414	0.747	1.030
	768	0.899	0.000	0.860	0.760

stable interpretability statistic.

F Appendix Showcase: Concept Steering

The steering examples in Section 5 are selected through a two-stage discovery-and-validation pipeline. We first shortlist candidate latents either by description search in a local feature browser or by low semantic entropy. These candidates are then reviewed using top activations, near-miss examples, and multiple description styles (old FADE-style, contrastive FADE-style, and Neuronpedia-style) so that unstable cluster captions can be filtered out before any causal claim is made. This extra review matters in practice: some latents remain semantically stable across description methods, whereas others vary enough across prompts that they should not be treated as clean showcase features.

F.1 Discovery by Semantic Entropy

The low-semantic-entropy branch of this discovery pipeline is meant to prioritize latents whose strongest token responses already form a semantically coherent cluster before any manual prompt inspection. While prior work has proposed sentence-level measures of semantic uncertainty (Kuhn et al., 2023; Farquhar et al., 2024), we adapt the same intuition to single hidden dimensions and use it as a neuron-discovery heuristic rather than as a standalone model-comparison benchmark.

Let ℓ denote the layer index and k the neuron index. For each neuron (ℓ, k), we first collect the token set whose mean activation exceeds a threshold.

Table 16: Sweep summary relative to the matured dense $k = 1536$ model at 20k steps. These metrics quantify how close each TopK model remains to the dense reference at the output and block-representation levels.

TopK model	Logit cos.	Top-1 agr.	Top-5 ovlp.	Block CKA	Block aligned cos.
$k = 64$	0.9281	0.7421	0.6663	0.6758	0.6038
$k = 128$	0.9354	0.7504	0.6819	0.7759	0.6607
$k = 256$	0.9342	0.7445	0.6807	0.7596	0.6409
$k = 512$	0.9388	0.7236	0.6727	0.7720	0.6516
$k = 768$	0.9403	0.7193	0.6716	0.8160	0.7274

Table 17: Continued-training summary for the deeper 24-layer $k = 64$ TopK model. Both 10k and 20k dense references are now available, so the main read is the within-horizon matched-FADE delta between TopK and dense for each layer band.

Variant	Val. Loss	CORE	Task Avg.	Early Δ	Middle Δ	Final Δ
Baseline 10k	**2.4434**	**0.2678**	**54.56**	—	—	—
Baseline 20k	**2.3913**	**0.2733**	**56.76**	—	—	—
$k = 64$ 10k	2.8272	0.1640	46.68	$+0.0353$	$+0.0721$	**+0.1083**
$k = 64$ 20k	2.7795	0.1852	46.76	$+0.0316$	**+0.1285**	$+0.0177$

Denote by $A^{(\ell)}_{k,d}$ the average activation of neuron (ℓ, k) to token d, and let θ be 70% of the 99.9th-percentile of all $A^{(\ell)}_{k,d}$ values. We then define

$$V_{\ell,k} = \{\, d \mid A^{(\ell)}_{k,d} > \theta \}. \tag{4}$$

To quantify how semantically tight this selected vocabulary subset is, we compute pairwise cosine similarity between token embeddings,

$$S_{d,d'} = \cos(e_d, e_{d'}) \quad \forall d, d' \in V_{\ell,k}, \tag{5}$$

weight each pair by corpus frequency,

$$w_{d,d'} = f_d \cdot f_{d'} \quad \forall d, d' \in V_{\ell,k}, \tag{6}$$

and bin the resulting similarities into a weighted histogram:

$$p_i = \sum_{(d,d') \in B_i} \frac{w_{d,d'}}{\sum_{d,d' \in V_{\ell,k}} w_{d,d'}}. \tag{7}$$

The semantic entropy of neuron (ℓ, k) is then

$$H_{\text{sem}}(\ell, k) = -\sum_{i=1}^{n_{\text{bins}}} p_i \log_2 p_i, \tag{8}$$

with $n_{\text{bins}} = 1000$ in our experiments. Low values indicate that the neuron's strongest token responses concentrate on a narrow and semantically coherent token set, so these neurons are natural candidates for further qualitative review. In practice, low semantic entropy is only a first-pass filter: we still inspect top activations, near misses, and multiple description styles before deciding whether a latent is stable enough to use in the steering showcase.

For the causal step, we steer one latent at a time with a `clamp_min` intervention, replacing its activation with $x \leftarrow \max(x, f)$ at the target layer. Rather than using a single global factor, we sweep factors derived from each latent's observed activation distribution, including median-like operating points and stronger tail values. The main prompt set is a small bank of generic prefixes (for example "Once upon a time," "In general," "I think," "For example," and "It is important to note that") chosen to avoid strongly cueing a target concept before intervention. Each candidate is then evaluated with two readouts:

1. generated continuations, which provide an intuitive qualitative check but can fail at large factors because of repetition or degeneration; and
2. next-token lexicon mass, which measures whether the intervention actually increases probability mass on a hand-specified concept lexicon.

We treat a latent as suitable for the qualitative showcase only when these two views are at least broadly consistent.

Table 18 first gives the complete representative prompt sweep for the main steering case $L21{:}412$, so that the paper's central study / publication latent can be inspected beyond the single main-text prompt. Tables 19 to 21 then extend the same

prompt-sweep view to the three neighboring main-family cases that support the same reading at adjacent layers and discourse roles. Tables 22 to 29 then give one table per appendix-side positive feature so that prompt robustness can be read feature-by-feature rather than only through a pooled showcase table. $L20{:}1250$ is the closest to the main study / publication family and usefully shows that neighboring discourse features can steer toward article-title or article-summary framing even when they are not as clean as $L21{:}412$. $L09{:}464$ then provides a distinct journalistic or attribution-style discourse case, while $L13{:}972$ adds a commercial or service-promotion family that is clearly separable from the paper-facing study / publication family. $L20{:}399$ then gives a lexical date-expression case. $L22{:}824$, $L22{:}315$, $L22{:}248$, and $L21{:}1284$ broaden coverage with topical or lexicalized continuations around bees, biblical language, shoes, and photography. The structural control set $L21{:}20$, $L22{:}79$, and $L22{:}752$ remains useful in the background because these latents still move the continuation but often do so through brittle punctuation, repetition, or locally lexicalized shifts rather than through a robust paper-facing semantic family. Together, these appendix cases explain why we make $L21{:}412$, $L17{:}791$, $L11{:}1403$, and $L18{:}546$ the main text: the model contains a wider variety of steerable latents, but the article / study / publication family remains the cleanest and most defensible showcase.

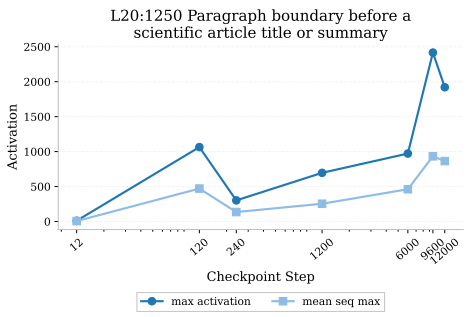
G Appendix Showcase: Tracing Feature Formation Across Training Checkpoints and Layers

To trace how individual neurons acquire concept-specific selectivity, we compare their activation statistics both across training checkpoints and along the depth of the network. The appendix figures focus on feature-level evidence rather than older aggregate entropy summaries: checkpoint-wise formation plots show when a latent first becomes semantically readable over training, and cross-layer plots show how nearby sparse dimensions differentiate into related but non-identical discourse or topical roles across depth.

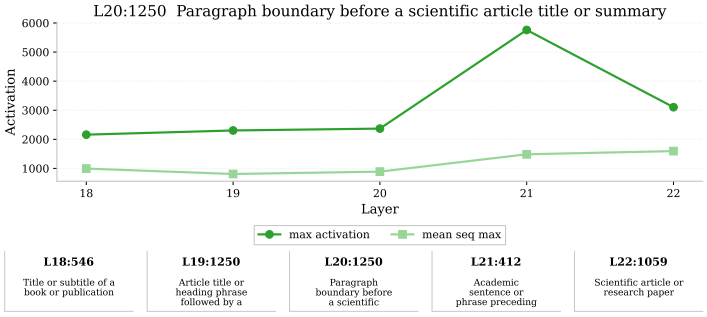
We further investigate two central questions about this formation process: (1) at which stage of training do successful features first become semantically readable, and (2) whether the same semantic family appears identically across layers or instead differentiates into depth-specific variants. This is one of the clearest places where the TopK design itself matters. Unlike dense-baseline analyses based on post-hoc SAEs, we do not need to train a separate feature dictionary for every checkpoint and every layer and then solve a cross-dictionary alignment problem before asking whether the "same" feature persisted. Checkpoint-wise, the TopK-based framework lets us track the same hidden-dimension index exactly across training. Layer-wise, it gives us an explicit sparse latent basis in each layer, so we can compare semantically related candidates across depth, but this second comparison is a semantic-alignment analysis rather than exact same-dimension tracing.

In the main text we therefore center the tracing story on the study / publication family, where this distinction is easiest to read. $L21{:}412$ provides the cleanest single checkpoint-wise and layer-wise case, while nearby $L17{:}791$, $L11{:}1403$, and $L18{:}546$ show that related discourse roles emerge across adjacent layers rather than as identical copies. The appendix-side positive cases widen semantic coverage, but they are mostly better read as supplementary steering, formation, or cross-layer evidence than as equally strong formation case studies. In particular, $L20{:}1250$ remains closest to the main family, whereas $L20{:}399$, $L22{:}824$, $L22{:}315$, and $L22{:}248$ are more lexical, topical, or stylistically niche and therefore less central to the paper's tracing claim. Supplementary formation and cross-layer examples for these appendix-side cases are collected in Figure 6.

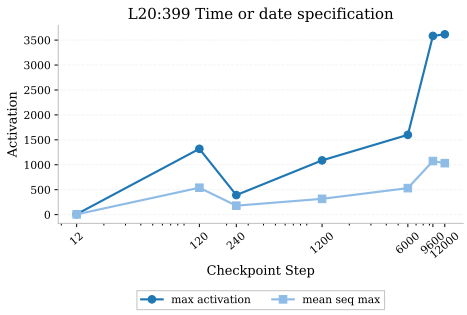
The structural controls reinforce the same point from the opposite direction. Latents such as $L21{:}20$, $L22{:}79$, $L21{:}1284$, and $L22{:}752$ can be activated and can visibly perturb the continuation, but they do not give comparably clean evidence of stable paper-facing feature formation. The key point is therefore twofold: checkpoint-wise tracing is exact because the same layer:latent can be followed directly over training, whereas layer-wise comparison is a semantic-alignment analysis within the model's explicit sparse bases. This is exactly why we treat feature formation and steering as supportive evidence: the 24-layer model does learn stable, human-readable sparse features, but their developmental timing, semantic breadth, and layer-wise role are heterogeneous rather than uniform.



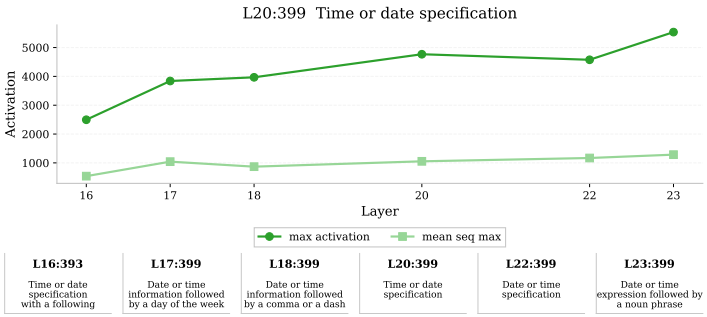
(a) $L20{:}1250$: checkpoint-wise formation



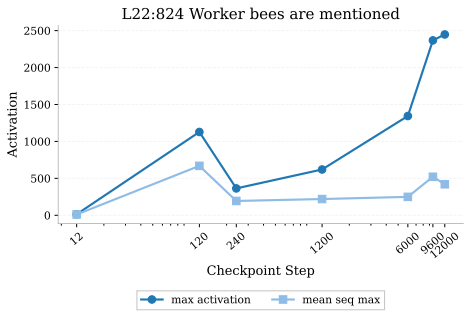
(b) $L20{:}1250$: layer-wise semantic alignment



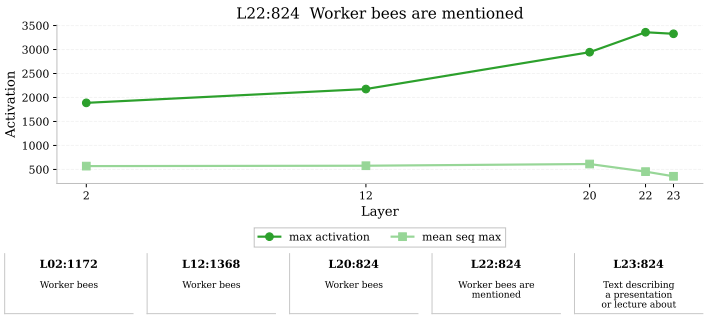
(c) $L20{:}399$: checkpoint-wise formation



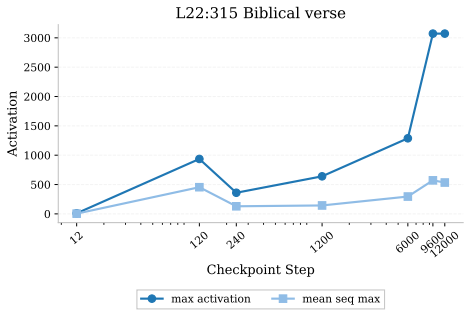
(d) $L20{:}399$: layer-wise semantic alignment



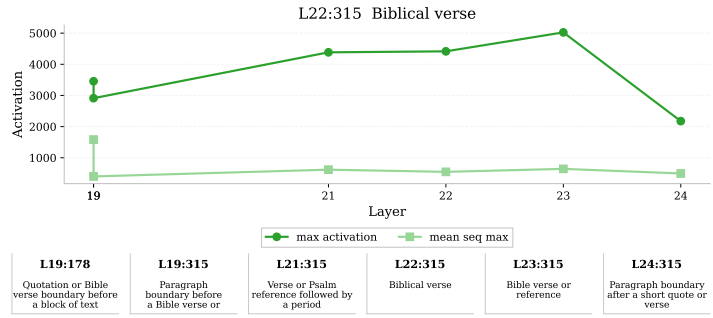
(e) $L22{:}824$: checkpoint-wise formation



(f) $L22{:}824$: layer-wise semantic alignment



(g) $L22{:}315$: checkpoint-wise formation



(h) $L22{:}315$: layer-wise semantic alignment

Figure 6: **Supplementary checkpoint-wise formation and layer-wise semantic alignment for appendix-side feature cases.** For each feature, the left panel traces the exact same layer:latent across checkpoints, while the right panel compares semantically related candidates across layers at the final checkpoint. $L20{:}1250$ remains the closest appendix-side case to the main study / publication family, but even there the cross-layer view separates article-title, article-summary, and related discourse-boundary roles rather than yielding identical copies. $L20{:}399$ shows a narrower lexical family built around date and time expressions. $L22{:}824$ shows that topical appendix cases can also become readable and align across layers, even though they are less central to the paper's main tracing claim than the study / publication family. $L22{:}315$ adds a more niche but semantically coherent biblical-language case, showing that the same appendix-side tracing view also applies to a strongly stylized discourse family.

Table 18: Prompt-wise steering for $L21{:}412$: academic sentence or phrase preceding a research study description.

Prompt	Baseline completion	Steered completion
For example,	the 1200 series of water heaters have an energy-efficiency rating of 12.5 . . .	the **study's research** found that when the researchers manipulated the brain's reward center . . . **This is a very interesting finding** . . .
I think	that a great way to get around that is to change your name . . .	that a great way to get **research** into the science and research community is to use **studies** that have shown a lot of **results** . . .
It is important to note that	the use of medications to treat pain should not be used as a substitute for professional medical advice . . .	the use of a specialized search engine like Nerpine can help you find your preferred types of **research**. **Research In The UK** . . . **the purpose of the study** . . .
In general,	it's not good practice to take your dog on walks in public places . . .	it's not good practice to take your dog on long walks in the summer . . .
Once upon a time,	in a land far away, there lived two best friends named Lily and Max . . .	in a small town named Harmville, lived four friends . . . They all had different stories to tell . . .

Table 19: Prompt-wise steering for $L17{:}791$: academic or research study description.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the following is a list of important features of the 2017–2018 **Health and Nutrition in Older Adults Study (HNOSA)**. The **study** is a follow-up study . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best way to get rid of them is to use a drug like methadone . . . I was taking it for 6 months . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in humans is very different from the use in other animals . . . **researchers have found** that the use of antibiotics in . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the **study's results suggest** that the treatment had no effect on the overall survival of patients with aggressive leukemia. The **study**, published in the **Journal of the National Cancer Institute**, . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the "Animal Shelter" . . .	in a land filled with friendly animals, there was a magical building called the Animal Health Centers . . . including cats, dogs, hamsters, gerbils, rats, . . .

Table 20: Prompt-wise steering for $L11{:}1403$: article or research study summary.

Prompt	Baseline completion	Steered completion
For example,	the 1200 series of water heaters have an energy-efficiency rating of 12.5 . . .	the **new study** finds that even when the virus has mutated, it is still capable of changing its course . . . according to a . . .
I think	that a great way to get around that is to change your name . . .	that a **new study** could also take a new look at the relationship between social distancing and **transmission** of the novel coronavirus in Michigan . . . **Nature Medicine** . . .
It is important to note that	the use of medications to treat pain should not be used as a substitute for professional medical advice . . .	the **new study** may not be the best study to use for a new study . . . **research studies** that are based on different study areas . . .
In general,	it's not good practice to take your dog on walks in public places . . .	it is not recommended to use artificial sweeteners, as a **new study** has found that people who drink artificial drinks at night may be more likely to experience memory loss . . . **Journal of Neuroscience** . . .
Once upon a time,	in a land far away, there lived two best friends named Lily and Max . . .	in a lush green forest, lived many different animals. A **new study finds** that new research shows that a key new species of bird has been found . . . **published in The Auk** . . .

Table 21: Prompt-wise steering for $L18{:}546$: title or subtitle of a book or publication.

Prompt	Baseline completion	Steered completion
For example,	the 1200 series of water heaters have an energy-efficiency rating of 12.5 . . .	the world's most comprehensive **study** of the causes of blindness and loss of vision was recently published . . . published in the November 15, 2013 issue of the journal, **Science** . . .
I think	that a great way to get around that is to change your name . . .	that a great way to get around that is to use the S3 as a tool rather than a tool for making a product . . .
It is important to note that	the use of medications to treat pain should not be used as a substitute for professional medical advice . . .	the use of the term "wallet" in this context is not a reference to the "wallet" used in the previous discussion . . . the context of the **article** . . .
In general,	it's not good practice to take your dog on walks in public places . . .	it's not good practice to take an extra step to make a transition smooth . . .
Once upon a time,	in a land far away, there lived two best friends named Lily and Max . . .	in a land far away, there lived two best friends named Lily and Max . . . they decided to build a tiny home in their backyard using just three materials . . .

Table 22: Prompt-wise steering for $L20{:}1250$: paragraph boundary before a scientific article title or summary.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the following is a list of important chemicals, drugs, and chemicals that help to maintain human health and safety. The **article** gives more detailed information . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . . (**To learn more about the benefits of fertilizer, I recommend this article**.) . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in humans is usually prohibited in animals . . . **The Role of Antibiotics in Diseases and Infections** . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the idea that a substance will undergo a chemical change in the presence of moisture is well-known . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the “Animal Shelter” . . .	in a bustling city lived two best friends named Sam and Alex. They were both curious about **science** and loved learning new things . . .

Table 23: Prompt-wise steering for $L09{:}464$: journalistic or news article quote or attribution.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the following is a statement on Wednesday, Nov. 14, 2013, in the **News Tribune**. “The United States Department of Health and Human Services” . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the new Halo game is very fun. The first time I played it was in the demo booth at IMDb’s show on Tuesday . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in animal production is prohibited in Europe . . . the **European Commission** has issued a warning on Friday . . . **EU News Service** . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the world’s population of 24 million is the world’s largest country . . . **Dalian, July 30 (AFP)** — the largest country in the world has announced . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the “Animal Shelter” . . .	in a bustling city filled with towering skyscrapers and whizzing traffic, there was a small but vibrant food truck known as “Vivi” . . .

Table 24: Prompt-wise steering for $L13{:}972$: continuous service promotion or description.

Prompt	Baseline completion	Steered completion
For example,	if we use the argument: “We have to use the same physics in the real world as in the virtual world” . . .	if we use the argument: “We have to use Uru engines to move in this market,” we’ll get the same answer . . . “We have to use **PPC** to deliver Uru engines to our target” . . .
I think	the idea that a well maintained car should not need a tune-up is too simplistic . . .	the idea that a user will have to pay for a **product** is a myth . . . make it clear to the user what the product is, and what it does . . .
It is important to note that	the following is just a brief overview of the various types of fish that can be found in the Gulf of Mexico . . .	the following is just a snippet from a website listing the top eight top online retailers that sell **sneakers** . . . research the specific brands and features . . .
In general,	the larger the group of people that you have in your group, the more likely you are to have a problem . . .	the larger the group of people that you have in your **network**, the more likely you are to have a success . . . your audience is more likely to be interested in your **product or service** . . .
Once upon a time,	in a land far away called South Korea, there was a magical place called the Kwangju Cultural Park . . .	in a land far away called **Cyberspace**, there lived many different people who worked together every day . . . a girl named Emma got a new job as an **app developer** . . .

Table 25: Prompt-wise steering for $L20{:}399$: time or date specification.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the **15th June 2011**, **7 June 2011** and **16 June 2011** . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best way to get rid of them is to use a 2–4 year old, with **10/12/15** small stones. (**Jan. 1, 2, 4, 23 December 2010**) . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in humans is usually prohibited in animals, as this can lead to the development of antibiotic-resistant bacteria . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the idea that a group of animals can move in a circle on its own is unlikely to be true. In fact, the **2 January 2010**, **11th October 2010** and **22 September 2010** . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the “Animal Shelter” . . .	in a bustling city lived two best friends named Sam and Alex . . . they decided to go on an adventure to find out how different countries compare when it comes to their pet . . .

Table 26: Prompt-wise steering for $L22{:}824$: worker bees are mentioned.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the **bee** has a very sophisticated immune system . . . the **honey bee wings** . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best way to get rid of them is to use a liquid insecticide . . . harmful if they come into contact with **bee food** . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in **beekeeping** is regulated . . . practices that **beekeepers** can use to manage their colonies . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the hive's inhabitants are solitary . . . the queen feeds the bees . . . the **worker bees** feed the bees by collecting pollen . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the "Animal Shelter" . . .	in a bustling city lived two best friends named Sam and Alex . . . everyone was like a **bee** . . .

Table 27: Prompt-wise steering for $L22{:}315$: biblical verse.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the following is a list of important features of the 1440-050A Motorcycle: **Size** . . . **Weight** . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best example of this is the recent **creationist book** . . . one of the **biblical prophets** was . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of non-hematoxiledimonic drugs, such as quinine, in children with malaria can increase the risk of thrombocytopaiphobia . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the idea that a particular place is more or less inextraordained with the rest of the world is not a well accepted paradigm . . . the **Bible** often interpreted the place . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the "Animal Shelter" . . .	there was a group of people who lived in a place called the **Church of Jesus Christ of Latter-day Saints** . . .

Table 28: Prompt-wise steering for $L22{:}248$: shoes or shoe-related phrases.

Prompt	Baseline completion	Steered completion
For example,	the following is a list of important features of the 2017–2018 BMW 3 Series Coupe . . .	the following is a list of important features of the 2017–2018 model year . . .
I think	the best way to get rid of them is to use a liquid fertilizer that your plant needs . . .	the best example of this is the recent USC football team, where the sole member of the team has to wear a pair of **shoes** . . .
It is important to note that	the use of antibiotics in humans is usually prohibited in animals . . .	the use of antibiotics in humans is very different from the use in other animals . . . when you are getting a fungal infection, a doctor may prescribe an antibiotic . . .
In general,	the idea that a particular area of the brain is responsible for a particular behavior can be applied to many situations . . .	the amount of carbon dioxide in the atmosphere is increasing at a rapid pace . . . a new study published in the journal Nature Communications . . .
Once upon a time,	in a land filled with friendly animals, there was a magical building called the "Animal Shelter" . . .	in a bustling city lived two best friends named Sam and Alex . . . how different types of **sneakers** can affect people's feet . . .

Table 29: Prompt-wise steering for $L21{:}1284$: paragraph boundary before a mention of photography or a photograph.

Prompt	Baseline completion	Steered completion
For example,	if we use the argument: "We have to use the same physics in the real world as in the virtual world" . . .	if we use the term "**photographer**" to describe a photographer, then your **photo** needs to be able to capture the essence of the scene . . .
I think	the idea that a well maintained car should not need a tune-up is too simplistic . . .	the idea that a **picture** alone can tell you everything you need to know about a person is a bit misleading . . . taking **pictures** and telling them the story . . .
It is important to note that	the following is just a brief overview of the various types of fish that can be found in the Gulf of Mexico . . .	the quality of your **photos** may vary depending on the specific camera model you use . . . capture stunning **photos and videos** . . .
In general,	the larger the group of people that you have in your group, the more likely you are to have a problem . . .	the larger the group of people in the room, the more likely it is that people are going to see it. If you're making a **portrait** . . . if you're making a landscape portrait . . .
Once upon a time,	in a land far away called South Korea, there was a magical place called the Kwangju Cultural Park . . .	in a faraway land called Cyprus, there lived many happy people who spent their days exploring the wonders of nature . . . "What do these **pictures** mean?" . . .