Conditional Dependence Structure in Sparse Autoencoder Features

Zachary Maas [1]

Abstract

Sparse Autoencoders (SAEs) decompose language-model activations into large overcomplete dictionaries of interpretable features, but these features are typically used as individual directions rather than as structured representations. We examine whether SAE features exhibit recoverable conditional dependence. We introduce a scalable estimator combining support-based screening, nodewise LASSO, and resampling-based null calibration to construct dataset-conditioned graphs over SAE features. Applied to GemmaScope SAEs on FineWeb and WMDP-bio, the graphs produced by this approach are sparse, stable under resampling, and modular, with many local neighborhoods linking semantically related features. These graphs also show shared cross-dataset and corpus-specific organization. They are not well explained by decoder cosine similarity or raw activation correlations. These results suggest that SAE representations contain organization beyond individual features, providing a way to study feature splitting, merging, and organization in overcomplete representations.

1. Introduction

Sparse autoencoders (SAEs) identify large collections of interpretable features in modern large language models (Bricken et al., 2023; Cunningham et al., 2023; Templeton et al., 2024; Lieberum et al., 2024). Despite their interpretability, the resulting representations lack clear organization. Empirically, these features also exhibit redundancy, including phenomena such as feature splitting and merging (Chanin et al., 2025). This behavior suggests the presence of latent organization governing interactions between individual features. Characterizing this latent structure may provide both practical and theoretical insight into SAE representations. Here, we introduce a structural view of SAEs by treating them as high-dimensional random vectors whose learned distribution induces latent interaction structure between features.

A natural approach to modeling interactions between SAE features is through dependence graphs (Lam, n.d.), identifying which features are predictive of one another. In classical settings, such structure is often approximated via the inverse covariance (precision) graph, using methods such as the graphical lasso (GLASSO) (Friedman et al., 2008). However, this is infeasible for SAEs in practice, since precision graph estimation scales with both the number of features and the number of data samples. SAEs have large numbers of features (10^5–10^7) and sparse activation structure, and broadly representative sampling from webtext-scale corpora is not feasible.

Recent work has increasingly used learned features for model monitoring, steering, and evaluation (Marks et al., 2025; Hubinger et al., 2024; Zou et al., 2025; Turner et al., 2024; Templeton et al., 2024; Lindsey et al., 2025). These approaches often rely on individual features. Other representation-learning approaches, including APD (Braun et al., 2025) and SPD (Bushnaq et al., 2025), may also induce dependence structure. We focus on SAEs here because they provide an overcomplete feature basis in which dependence structure between features is expected rather than incidental (Bricken et al., 2023; Olshausen & Field, 1997; Elad, 2010).

In this work, we introduce a scalable method for estimating conditional dependence graphs over SAE features. The approach combines three components. First, a support-guided sampling procedure restricts candidate features and edges using observed co-activation structure. Second, nodewise sparse regression (nodewise LASSO) (Tibshirani, 1996; Meinshausen & Bühlmann, 2006) is used to estimate local conditional dependencies in a computationally efficient manner. Third, resampling-based controls and null permutations are used to reduce false discoveries. Together, these components yield an approximate estimator of feature-dependence structure that is computationally tractable for SAEs.

Applied to SAEs trained on open large language models, the

[1]Independent, United States. Correspondence to: Zachary Maas <zach.maas@gmail.com>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

resulting graphs exhibit stable feature interaction structure under resampling. These graphs contain modular components that frequently correspond to coherent linguistic or semantic patterns. Across distinct datasets, the method recovers both clusters of corpus-specific behavior as well as cross-dataset shared structure. Moreover, the inferred graphs differ substantially from similarity-based baselines such as cosine similarity and activation correlation, indicating that the method captures structure not recovered by pairwise similarity alone.

This work contributes a computationally efficient method for estimating conditional dependence in sparse, high-dimensional feature representations, and provides empirical evidence for structured dependence between SAE features. **We show that SAE feature spaces contain recoverable higher-order organization that is not explained by pairwise similarity.** These results suggest a path toward organizing interpretable features into higher-level modules, with potential applications in both interpretability and downstream control.

2. Method

For notational convenience, let $X \in \mathbb{R}^d$ denote the vector of SAE feature activations from a fixed model under a given dataset. We define G_X as the latent conditional-dependence graph associated with that dataset. In practice, we construct an estimator $\hat{G}_X$ which we refer to as the dataset-conditioned graph.

At a high level, our approach approximates local dependence structure using three stages. First, we use sparse activation support to identify a restricted set of plausible candidate interactions between SAE features. Second, we fit sparse local regressions using nodewise LASSO to estimate approximate conditional dependence between those candidate features. Finally, we repeat this procedure across multiple reruns and retain only edges that recur consistently under resampling (Figure 1).

Rather than attempting global precision estimation over all SAE features simultaneously, our approach constructs a sparse approximate graph locally through screening, selective refinement, and stability selection. This substantially reduces the computational burden while still recovering stable dependence structure.

2.1. Data collection

We first collect sparse activation statistics needed for support-aware screening and downstream graph estimation.

Data collection consists of sampling token positions over a corpus while maintaining statistics required later in the pipeline. We draw a shared set of token positions from a selected dataset and evaluate all SAE layers on those same sampled rows. For each layer, we store sparse sampled-row activations and compute per-feature support summaries, including support count, activation mass, squared activation mass, and effective sample size.

Note that we share sampling across SAE layers for computational efficiency, but graph estimation itself remains strictly layer-local under this approach (Appendix A).

2.2. Single-fold graph estimation

We next estimate a single approximate dependence graph from one sampled fold.

For a fixed layer, we split sampled rows into disjoint screening and regression subsets. Let c_j denote the sampled support count of feature j, and let $u_j = |C_j^{(0)}|$ denote the number of candidate neighbors identified during a coarse screening pass on the screening split.

We first remove features with insufficient support or trivially small candidate neighborhoods before expensive refinement. We define the support-aware target set

$$A_{B,m} = \{j : c_j \geq B,\ u_j \geq m\},$$

where B is a minimum target-support threshold and m is a minimum coarse-candidate-count threshold (subsection A.1).

Even after screening, the number of eligible targets may still be computationally prohibitive. To concentrate compute on the most ambiguous and information-dense regions of the graph, we order targets in $A_{B,m}$ by screening evidence and retain the smallest prefix $T_{B,m,p}$ whose cumulative support mass covers a fraction p of the eligible ambiguous support mass:

$$\sum_{j \in T_{B,m,p}} c_j \geq p \sum_{j \in A_{B,m}} c_j.$$

We additionally apply a global cap $|T_{B,m,p}| \leq M_{\max}$ to bound the number of target features selected for selective resampling.

For each retained target $j \in T_{B,m,p}$, we refine candidate neighborhoods through repeated screening on row subsamples. This produces estimated candidate inclusion frequencies $\hat{\pi}_{jk}$. We then select a per-run inclusion threshold τ as the largest value preserving at least a ρ fraction of support-weighted candidate-union row coverage, and define the refined candidate neighborhood

$$\hat{C}_j = \{k : \hat{\pi}_{jk} \geq \tau\}.$$

We define U_j as the regression rows on which either the target feature j or at least one candidate feature $k \in \hat{C}_j$ is active.

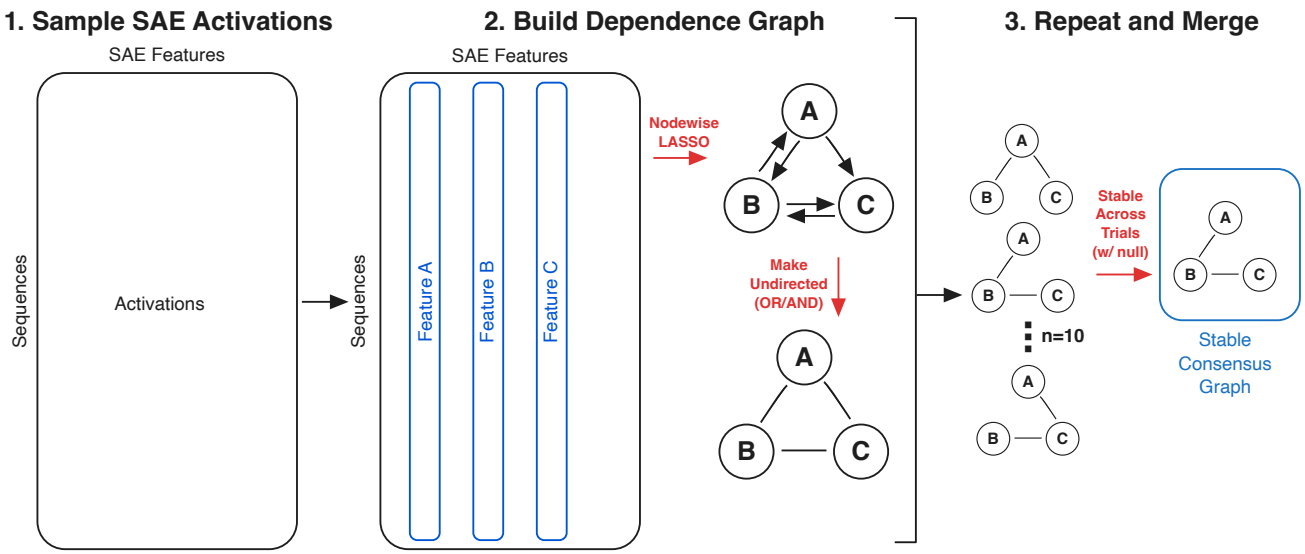

Figure 1. Overview of the support-aware Graph SAE pipeline. Given sampled SAE activations, we first use support-aware screening to restrict each target feature to a small set of plausible candidate neighbors. We then fit sparse nodewise LASSO regressions on candidate data and symmetrize the resulting directed coefficients into a per-trial approximate dependence graph. Repeating this procedure across resampled trials and feature-permuted nulls yields a stable graph of SAE feature dependencies.

Given a globally calibrated penalty λ identified with binary search (subsection A.2), we then fit the nodewise LASSO

$$\hat{\beta}^{(j)} = \arg\min_{\beta} \frac{1}{2|U_j|} \|x_j - X_{\hat{C}_j}\beta\|_2^2 + \lambda\|\beta\|_1.$$

We add a directed edge $k \to j$ whenever $\hat{\beta}_k^{(j)} \neq 0$. We form the per-fold undirected graph through OR symmetrization across directed edges by default, although we additionally evaluate stricter AND symmetrization later in Section 3.

2.3. Stable graph selection

Finally, we retain only edges that recur consistently across reruns.

We repeat the single-fold graph construction procedure across R observed folds and count how often each undirected edge appears. For an edge e, its observed recurrence count is

$$C_e^{\text{obs}} = \sum_{r=1}^{R} \mathbf{1}[e \in \hat{G}_r].$$

For each recurrence threshold k, we define the observed graph size

$$N^{\text{obs}}(k) = |\{e : C_e^{\text{obs}} \geq k\}|.$$

To calibrate false discoveries, we generate null folds by independently permuting each feature column over sampled activations while preserving marginal feature distributions. We then rerun the same screening and graph-estimation pipeline on these null activations. This yields null recurrence counts and corresponding null graph sizes $N_{\text{null},q}(k)$ across Q null repetitions.

We estimate false discovery rate at threshold k by

$$\widehat{\text{FDR}}(k) = \min\left(1, \frac{\frac{1}{Q}\sum_{q=1}^{Q} N_{\text{null},q}(k)}{\max(1, N^{\text{obs}}(k))}\right).$$

We select the smallest recurrence threshold $k^\star$ such that

$$\widehat{\text{FDR}}(k^\star) \leq \alpha.$$

The final stable graph retains all edges with observed recurrence in at least $k^\star$ trials. For multiple SAEs across distinct layers, we repeat this procedure independently for each SAE.

3. Results

We evaluate the full pipeline on FineWeb (Penedo et al., 2024) and WMDP-bio (Li et al., 2024) using Gemma 2 27b (Gemma Team, 2024) with Gemma Scope SAEs (Lieberum et al., 2024) at width 131k on all available layers (10, 22, and 34). FineWeb is used as a broad webtext corpus, and WMDP-bio is used to better understand domain-specific conditioning. Unless otherwise noted, all reported graphs use a fixed configuration of the described pipeline (Appendix A), with 10 repeat trials and 3 null permutations per trial. FineWeb runs use target-degree LASSO calibration, while WMDP-bio runs use a support-scaled penalty

because the smaller domain-specific corpus shows substantially more heterogeneous target support (subsection A.2). Across these runs and datasets, our pipeline recovers dependence graphs that exhibit four consistent properties:

1. They are large, sparse, and modular, containing subgraphs that link semantically related SAE features (Figure 2).
2. They contain stable backbone structure that persists under resampling and, in part, across datasets.
3. They recover dataset-specific communities that reflect the underlying corpus.
4. They are not well explained by decoder cosine similarity or raw activation correlation, indicating that the estimator captures structure beyond simple pairwise similarity.

Additionally, we show that our pipeline recovers compositional structure in a synthetic oracle setting (subsection A.4).

3.1. Graph Structure

Our methodology produces robust graphs that span a substantial fraction of the 131k SAE features. The generated graphs are too large to effectively present here, and are provided in the supplementary material.

On FineWeb (Table 1), we see notable variation in graph structure across layers with OR symmetrization. Layer 10 consists of a single large connected component with only moderate modularity and relatively few identified communities. At layer 22, we see substantial contraction in the graph size as well as fragmentation of the graph into many smaller components. By layer 34, we observe a jump in modularity suggesting stronger community separation. In contrast, the graphs identified for WMDP-bio are substantially smaller and show greater fragmentation into separate communities. This likely reflects the substantially lower textual diversity of WMDP-bio relative to the broad webtext corpus in FineWeb.

To better understand the highest confidence core of these graphs, we perform a stricter secondary comparison using AND symmetrization to keep only bidirectional edges (Table 1). As expected, this shrinks the resulting graphs dramatically, while maintaining a strict subset of the original OR symmetrized graph. This choice yields a qualitatively different graph object. Instead of estimating broad approximate dependence structure, AND symmetrization isolates only reciprocal dependencies between features. This is a much stricter condition, but may yield a more conservative approximation of G_X depending on the context. For the purposes of this work, we consider the OR symmetrization to be our primary graph estimator, yielding a larger and richer graph for downstream analysis. Across these graphs, we see that based on automatically generated labels, our identified node/edge pairs have higher semantic similarity than expected under a label-shuffle null (subsection A.3).

3.2. Shared Backbone and Dataset-Specific Structure

Our approach recovers both cross-dataset shared structure as well as dataset-specific structure. Between the graphs estimated on FineWeb and WMDP-bio, we observe $2{,}517$ shared stable edges and $1{,}753$ nodes incident to those shared edges at layer 10. At layer 34, the overlap is smaller but still nontrivial, with 536 shared stable edges involving 470 nodes. Inspection of these shared subgraphs indicates that they primarily reflect lower-level structural scaffolding, including technical-description and formatting patterns (Figure 3).

The existence of overlap suggests our approach is capturing shared elements across dataset conditioning. Overlapping nodes and edges between datasets are expected in this context, since natural language corpora (particularly those constrained to a single language) share basic statistical structure that should be reflected if present in the SAE's underlying dependence structure. At the same time, while we observe shared cross-dataset structure, in general it appears that the organization of the graphs generated by our method tends to reflect domain-specific characteristics of the dataset used.

The choice of WMDP-bio as a validation dataset for this work was purposeful — with this dataset, SAEs have previously been shown to be useful towards safety ends for unlearning biosafety capabilities via both single and multiple SAE features (Farrell et al., 2024). The dominant communities from this dataset are more domain-specific than on FineWeb, including a variety of features related to human-health and virology-related clusters. Dataset specificity supports the claim that the approach described here can produce conditional dependence structure that is specifically conditioned on the input dataset.

In order to further test the dataset conditioning of graphs, we explored the stability of the output graphs in response to superficial prompt-formatting changes. In a matched control, where we only alter dataset QA scaffolding, all three layers retain the same $k^\star = 2$ stability and only modestly reduce their stable-edge counts. As an example, layer 34 is reduced in size from $20{,}042$ to $18{,}817$ stable edges and layer 10 from $22{,}009$ to $21{,}023$. Across layers, 23–35% of stable edges from these runs are shared, and the primary identified communities remain biology and virology focused. However, this also suggests that these graphs are nonetheless relatively sensitive to dataset formatting when using a constrained format for input (as opposed to general webtext).

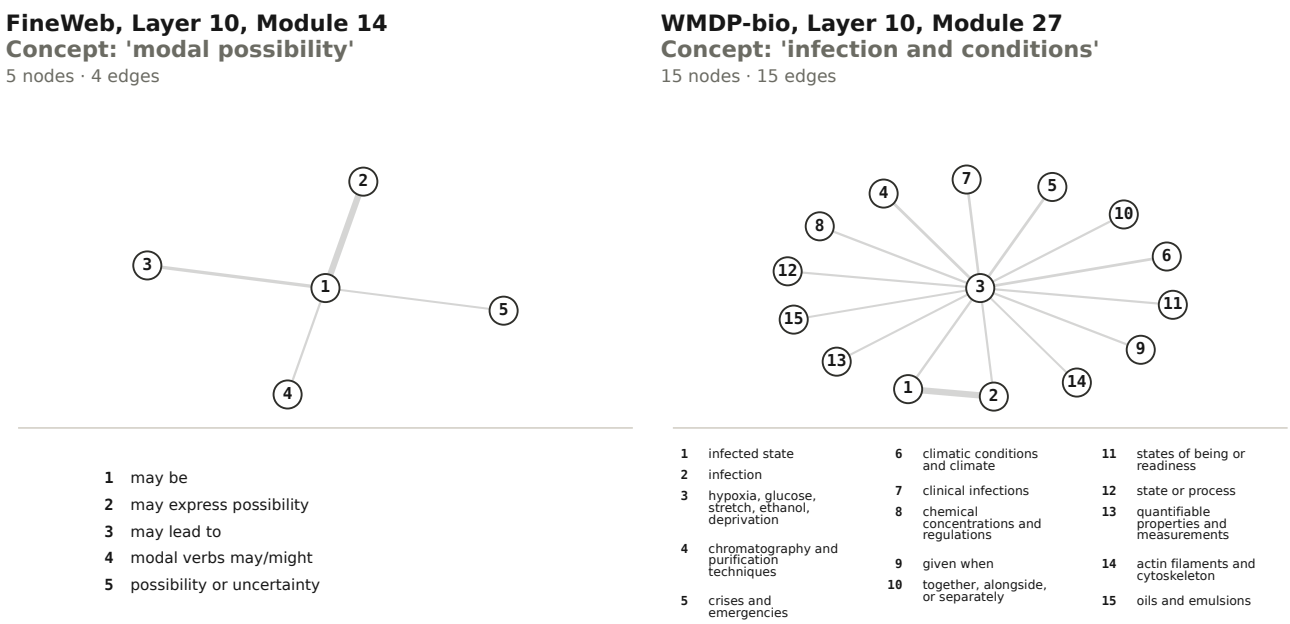

Figure 2. Small sampled Louvain modules from the full graphs for FineWeb and WMDP-bio at layer 10. Most identified modules are larger, but for visualization of semantic relation between features we have selected two samples that are small enough to display here. Note that the selected modules show hub-and-spoke structure but this is not universal to all identified modules. We highlight these modules to support the notion that identified clusters and modules from these graphs can be semantically coherent, although this is not universally the case. Node labels are Neuronpedia (Lin, 2023) feature labels. **Left:** a small cluster from our generated FineWeb graph showing shared concepts around "modal possibility", linking various uses of the word "may" in a context. **Right:** a slightly larger cluster from our generated WMDP-bio graph containing more varied concepts, which appear to capture concepts relating to infection, biological conditions, and health related states.

3.3. Validation

The recovered graphs exhibit stable recurrence under resampling and frequently contain semantically coherent local structure. An important remaining question is whether this organization reflects genuinely higher-order structure or can instead be explained by simpler pairwise similarity measures. We test against these baselines and find that pairwise similarity appears insufficient to recover the estimated graph structure.

We compare the stable LASSO graph against two pairwise-similarity baselines. First, for each layer, we build a sparsity-matched baseline by taking the same node set as the LASSO graph and selecting exactly as many edges as the LASSO graph contains, choosing the highest-scoring decoder-cosine or raw activation-correlation pairs globally. Second, we assess a stronger baseline using the same node set and per-node degree as the LASSO graph, choosing cosine-similarity or correlation scored edges within that constraint. Across the FineWeb graph at layers 10, 22, and 34, the same-sparsity cosine graph recovers only 0.77% to 1.91% of stable LASSO edges, while same-sparsity raw correlation recovers 0.00% to 1.46%. **Even degree-matched similarity graphs recover only 1.1%–5.5% of stable LASSO edges**, while correlation recovers 2.6% to 6.0%.

Succinctly, **degree-matched pairwise similarity graphs do not recover conditional structure**. These baselines indicate that our estimated graphs $\hat{G}_X$ are not well explained by either pairwise decoder feature similarity or raw co-activations. Taken together, these results indicate that our estimator is identifying novel organization that is not trivially explained by either decoder similarity or co-activation alone (Table 2).

To further evaluate whether the graph-estimation procedure can recover known compositional structure under controlled conditions, we additionally evaluated the method in a synthetic oracle setting where the relevant latent variables were directly available or provided through a fixed oracle representation (subsection A.4). In these settings, the procedure recovered compositional motifs with high recall, while cosine-similarity baselines failed to recover the same structure.

Across runs, null controls typically identify a consensus threshold for graph convergence in the range of $k^\star = 2$–4 (null-estimated survival FDR below $q = 0.1$), where an edge is considered to be in the consensus set if it recurs in

Dataset	Layer	Symmetrization	k^*	Nodes	Edges	CCs	LCC	Q	Comms.
FineWeb	10	OR	3	19,900	73,598	1	100.0%	0.692	14
FineWeb	10	AND	2	606	3,063	4	98.8%	0.588	11
FineWeb	22	OR	3	7,257	11,825	9	99.7%	0.709	20
FineWeb	22	AND	3	84	71	24	16.7%	0.157	26
FineWeb	34	OR	3	9,698	18,025	7	99.8%	0.850	27
FineWeb	34	AND	2	315	891	10	93.3%	0.557	19
WMDP-bio	10	OR	2	7,457	22,009	2	100.0%	0.787	31
WMDP-bio	10	AND	2	866	1,926	21	95.0%	0.835	41
WMDP-bio	22	OR	2	5,263	9,727	20	99.1%	0.826	57
WMDP-bio	22	AND	2	386	563	32	81.6%	0.815	48
WMDP-bio	34	OR	2	6,056	20,042	4	99.9%	0.720	23
WMDP-bio	34	AND	2	854	2,776	6	98.6%	0.720	25

Table 1. Stable graph structure under matched 'OR' and 'AND' symmetrization. 'CCs' denotes connected components, 'LCC' the largest connected-component share of nodes, Q Louvain modularity, and 'Comms.' the number of Louvain communities. Reciprocal 'AND' symmetrization yields a strict subset of the corresponding 'OR' graph

Layer	Graph	Edges	Shared w/ LASSO	Shared frac.	Jaccard	Components	LCC frac.
Layer 10	Stable LASSO	73,598	73,598	1.000	1.0000	1	1.000
Layer 10	Decoder cosine	73,598	3,394	0.046	0.0236	2,449	0.733
Layer 10	Raw activation corr.	73,598	4,197	0.057	0.0293	2,632	0.714
Layer 34	Stable LASSO	18,025	18,025	1.000	1.0000	7	0.998
Layer 34	Decoder cosine	18,025	399	0.022	0.0112	2,324	0.473
Layer 34	Raw activation corr.	18,025	546	0.030	0.0154	2,393	0.457

Table 2. Degree-matched similarity baselines for the full FineWeb k^* stable graph. Decoder-cosine and raw activation-correlation graphs are built on the same stable-node set and matched to the LASSO graph's degree sequence and edge count. Even under this favorable matching, simple similarity graphs share only a small fraction of the stable LASSO edges and fragment into many more components.

at least $k^\star$ trials. As k increases, the number of retained nodes and edges falls off rapidly (Figure 4). We do not interpret the model-selected $k^\star$ graph $\hat{G}_{k^\star}$ as a uniquely correct subset of G_X, but rather as a stable consensus over multiple estimated $\hat{G}_r$.

Empirically, increasing k causes the graphs to collapse toward small disconnected components dominated by formatting and general language structure. An arbitrarily strict k or q can be chosen depending on the intended downstream use of the graph. Because our interest is in identifying clusters of SAE features that may be useful for downstream applications, we bias selection towards retaining a larger set of candidates.

4. Conclusion

We present a scalable pipeline for estimating dataset-conditioned conditional dependence structure between SAE features. Using support-aware sampling, nodewise LASSO, and resampling-based calibration, our approach constructs sparse dependence graphs over large SAEs that are stable under resampling and recover both shared and dataset-specific organization. These graphs frequently contain edges and modules linking semantically related features, while also recovering structure not well explained by decoder cosine similarity or raw activation correlation. Together, these results suggest that SAE representations contain recoverable organization beyond individual feature directions alone.

This work introduces a structural view of SAEs by treating feature activations as high-dimensional random vectors whose learned distribution induces latent dependence structure. Under this framing, the graphs estimated here can be viewed as finite, dataset-conditioned approximations to a hypothetical latent precision graph associated with the SAE. In practice, such a graph is not directly recoverable due to the scale of modern SAEs and the infeasibility of sampling broadly from the model's learned distribution. Instead, any practical estimator must depend on both the dataset used and the estimation procedure itself. While we focus on nodewise LASSO due to its computational tractability and scalability, other sparse graphical estimators or nonlinear dependence measures may provide complementary views

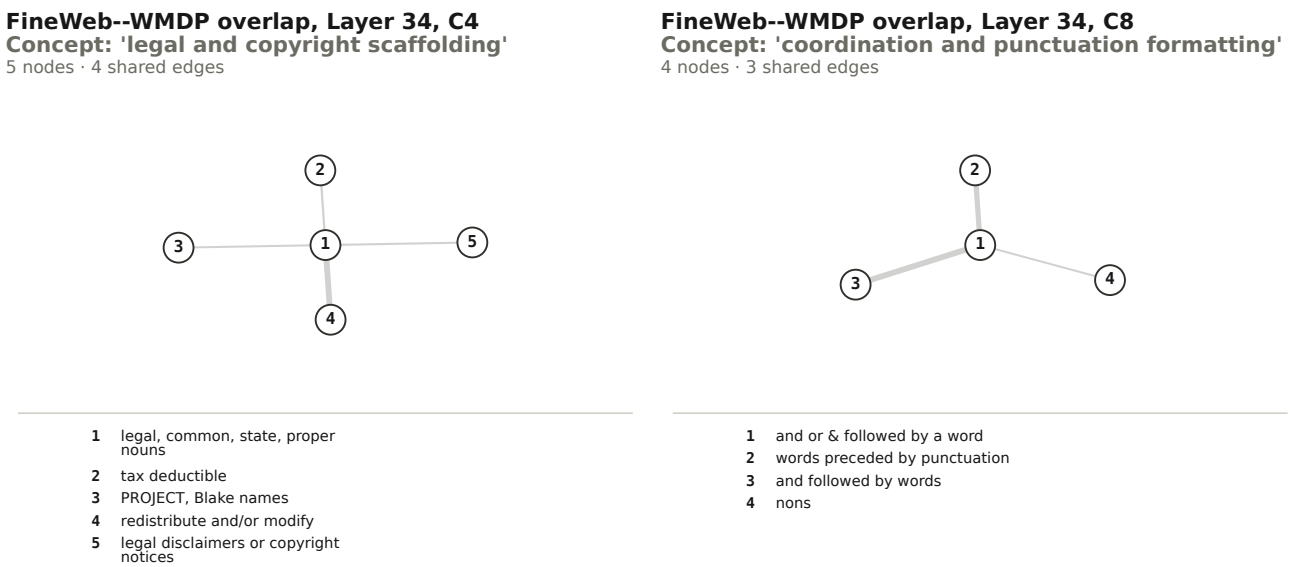

Figure 3. Shared FineWeb and WMDP-bio overlap components from layer 34. **Left:** An overlapping cluster of features containing features describing various legal concepts. **Right:** An overlapping cluster of features containing features related to textual conjunction and punctuation associated with it.

of SAE organization.

Our work has several limitations. The recovered graphs are not causal objects and should not be interpreted as mechanistic circuits in the classical sense. Rather, they capture statistical dependence between SAE features under dataset conditioning. This dependence may reflect shared computation, compositional structure, redundancy, or feature splitting within overcomplete representations. Notably, although many recovered modules are semantically coherent, this is not universally the case. The estimator frequently identifies stable structure that is not obviously interpretable at the semantic level, yet also is not recovered by similarity-based baselines. Semantic coherence is not the only meaningful organizing principle present in SAE representations. Additionally, we only examine a single model and SAE family here to validate our approach, and future work should consider additional models.

Several directions remain open for future work. An important next step is to better understand the relationship between feature-interaction structure and causal mechanisms in language models. It will also be important to study how these graphs vary across datasets, model families, and representation-learning methods. Finally, the non-semantic structure recovered by this approach may provide additional insight into how representations and computations are distributed across features in overcomplete systems.

Overall, our results suggest that SAE representations contain higher-order structure that can be systematically probed at scale. Characterizing this structure provides a useful foundation for organizing learned representations into larger functional modules and may support new approaches to monitoring and steering model behavior through interactions between features rather than isolated components.

References

Braun, D., Bushnaq, L., Heimersheim, S., Mendel, J., and Sharkey, L. Interpretability in Parameter Space: Minimizing Mechanistic Description Length with Attribution-based Parameter Decomposition, 2025. URL http://arxiv.org/abs/2501.14926.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Transformer Circuits, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

Bushnaq, L., Braun, D., and Sharkey, L. Stochastic Parameter Decomposition, 2025. URL http://arxiv.org/abs/2506.20790.

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., and Bloom, J. A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders, 2025. URL http://arxiv.org/abs/2409.14507.

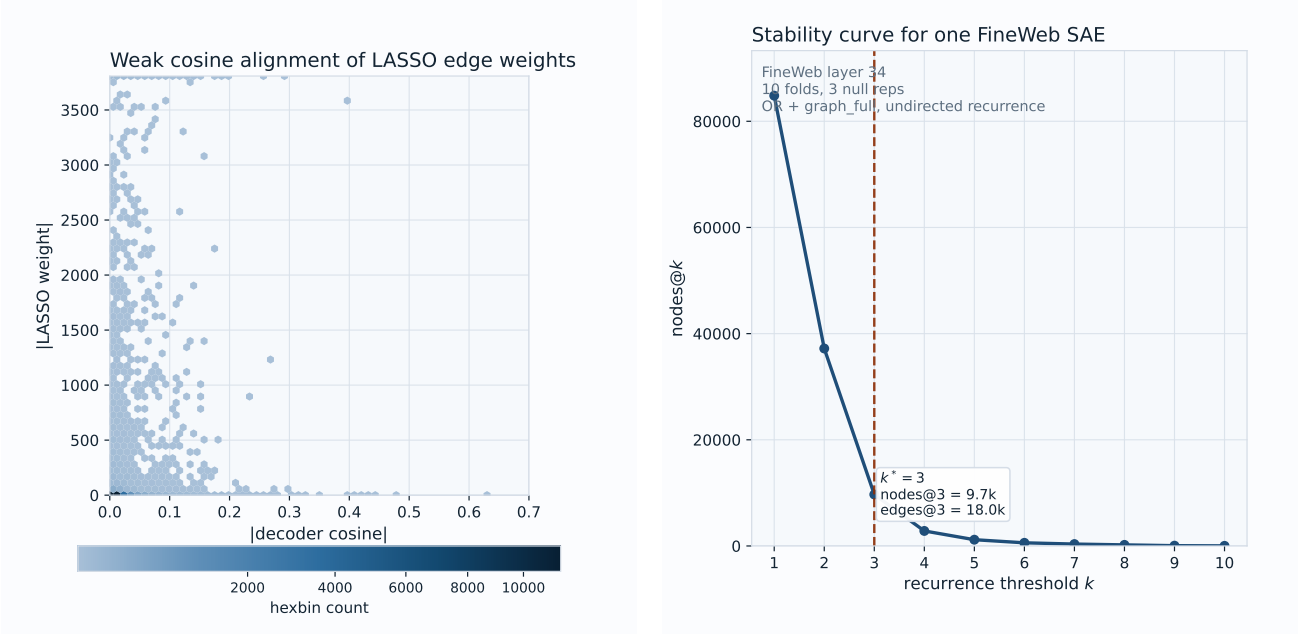

Figure 4. **Left:** decoder cosine similarity versus absolute LASSO weight for Gemma 2 27b, FineWeb, Gemma Scope layer 34. The weak alignment indicates that the recovered graph is not well explained by similarity between decoder vectors alone. **Right:** a stability curve for the same run, plotted as nodes@k across recurrence thresholds. With null calibration, $k^\star$ is typically in the range 2–4, which we treat as the consensus regime for this work.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse Autoencoders Find Highly Interpretable Features in Language Models, 2023. URL http://arxiv.org/abs/2309.08600.

Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer New York, 2010. ISBN 978-1-4419-7010-7 978-1-4419-7011-4. doi: 10.1007/978-1-4419-7011-4. URL https://link.springer.com/10.1007/978-1-4419-7011-4.

Farrell, E., Lau, Y.-T., and Conmy, A. Applying sparse autoencoders to unlearn knowledge in language models, 2024. URL http://arxiv.org/abs/2410.19278.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045. URL https://doi.org/10.1093/biostatistics/kxm045.

Gemma Team. Gemma 2: Improving Open Language Models at a Practical Size. Preprint, 2024. URL https://arxiv.org/abs/2408.00118v3.

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., Jermyn, A., Askell, A., Radhakrishnan, A., Anil, C., Duvenaud, D., Ganguli, D., Barez, F., Clark, J., Ndousse, K., Sachan, K., Sellitto, M., Sharma, M., DasSarma, N., Grosse, R., Kravec, S., Bai, Y., Witten, Z., Favaro, M., Brauner, J., Karnofsky, H., Christiano, P., Bowman, S. R., Graham, L., Kaplan, J., Mindermann, S., Greenblatt, R., Shlegeris, B., Schiefer, N., and Perez, E. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training, 2024. URL http://arxiv.org/abs/2401.05566.

Lam, C. High Dimensional Covariance Matrix Estimation, n.d. Manuscript.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Khoja, A., Zhao, Z., Herbert-Voss, A., Breuer, C. B., Marks, S., Patel, O., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Kaplan, R., Steneker, I., Campbell, D., Jokubaitis, B., Levinson, A., Wang, J., Qian, W., Karmakar, K. K., Basart, S., Fitz, S., Levine, M., Kumaraguru, P., Tupakula, U., Varadharajan, V., Wang, R., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning, 2024. URL http://arxiv.org/abs/2403.03218.

Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramár, J., Dragan, A., Shah, R., and Nanda, N. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2. Preprint, 2024. URL https://arxiv.org/abs/2408.05147v2.

Lin, J. Neuronpedia: Interactive Reference and Tooling for Analyzing Neural Networks, 2023. URL https://www.neuronpedia.org.

Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., Citro, C., Abrahams, D., Carter, S., Hosmer, B., Marcus, J., Sklar, M., Templeton, A., Bricken, T., McDougall, C., Cunningham, H., Henighan, T., Jermyn, A., Jones, A., Persic, A., Qi, Z., Thompson, T. B., Zimmerman, S., Rivoire, K., Conerly, T., Olah, C., and March 27, J. B. A. A. P. On the Biology of a Large Language Model, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Marks, S., Treutlein, J., Bricken, T., Lindsey, J., Marcus, J., Mishra-Sharma, S., Ziegler, D., Ameisen, E., Batson, J., Belonax, T., Bowman, S. R., Carter, S., Chen, B., Cunningham, H., Denison, C., Dietz, F., Golechha, S., Khan, A., Kirchner, J., Leike, J., Meek, A., Nishimura-Gasparian, K., Ong, E., Olah, C., Pearce, A., Roger, F., Salle, J., Shih, A., Tong, M., Thomas, D., Rivoire, K., Jermyn, A., MacDiarmid, M., Henighan, T., and Hubinger, E. Auditing language models for hidden objectives, 2025. URL http://arxiv.org/abs/2503.10965.

Meinshausen, N. and Bühlmann, P. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006. ISSN 0090-5364, 2168-8966. doi: 10.1214/009053606000000281. URL https://projecteuclid.org/journals/annals-of-statistics/volume-34/issue-3/High-dimensional-graphs-and-variable-selection-with-the-Lasso/10.1214/009053606000000281.full.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. ISSN 0042-6989. doi: 10.1016/S0042-6989(97)00169-7. URL https://www.sciencedirect.com/science/article/pii/S0042698997001697.

Penedo, G., Kydlíček, H., allal, L. B., Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. V., and Wolf, T. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, 2024. URL http://arxiv.org/abs/2406.17557.

sentence-transformers. Sentence-transformers/all-MiniLM-L6-v2 · Hugging Face, 2024. URL https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. Transformer Circuits, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/.

Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246. URL https://www.jstor.org/stable/2346178.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering Language Models With Activation Engineering, 2024. URL http://arxiv.org/abs/2308.10248.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers, 2020. URL https://arxiv.org/abs/2002.10957v2.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., Kolter, J. Z., and Hendrycks, D. Representation Engineering: A Top-Down Approach to AI Transparency, 2025. URL http://arxiv.org/abs/2310.01405.

Appendices

A. Additional Pipeline Details

For replication, code is available at https://github.com/zmaas/graph_sae.

Below, we report additional implementation details omitted from the main text for brevity:

- When resampling data across trials, the folds are not strictly disjoint. Each stability trial uses an independently seeded set of sampled rows. In our FineWeb

layer-34 run, pairwise row-coordinate overlap is about 12.6%, matching independent-sampling expectation, and no coordinate is shared across all 10 folds.

- For computational efficiency when doing analysis across multiple folds, the same sampled token rows are reused across SAE layers 10/22/34 within a fold. Support profiling and graph estimation remain layer-local.
- An earlier version of the pipeline using fully independent random sampling (vs the support-aware method used here) used n=3 null repeats for stability. Our current method does not require this, but we have retained the n=3 parameter to keep null-calibration fixed across experiments. In a FineWeb layer-34 sensitivity check, r in $1, 2, 3, 5, 10$ all recovered the same $k* = 3$ and the same stable graph.
- Consensus is recurrence, not majority vote. With $(R = 10)$, $k* = 2$ means an edge appears in at least 2 observed reruns. $k* = 3$ means at least 3.
- We run the 27B base model with int8 weight quantization for memory efficiency, while keeping the SAE weights in their standard loaded precision. All downstream sparse activations, support profiling, screening, LASSO, and graph estimation use the resulting SAE activations.
- In initial development, the pipeline initially used 30-fold repeat trials. However, after identifying that $k^\star$ was always less than 5, we moved to 10-fold repeat trials for compute efficiency.
- While nodewise LASSO with OR-symmetrization has well-known recovery guarantees (Meinshausen & Bühlmann, 2006), SAE activations are typically sparse, heavy-tailed, and strongly correlated, all of which violate the usual Gaussian assumptions. We therefore do not interpret our recovered graph as a statistically-idealized estimate of a true precision graph, but rather a computationally tractable approximation in an empirical setting. As noted by (Meinshausen & Bühlmann, 2006), our selected estimator may remain useful even with deviations from Gaussianity.

A.1. Hyperparameter Selection

Our current implementation of this pipeline contains four compute-constraining hyperparameters that control compute:

- p**, support-mass coverage.** By default, $p = 0.99$. After applying the screening gates below, we greedily retain the smallest prefix of eligible target features whose cumulative support count covers at least a p fraction of the eligible ambiguous support mass.
- B**, minimum support.** A feature must appear in at least B sampled rows before it is eligible for selective resampling. This removes very low-support features from the expensive edge refinement stage.
- m**, minimum coarse-candidate count.** Among supported features, a feature must have at least m coarse screening candidates before it is selected for possible resampling. This focuses refinement on targets with nontrivial candidate neighborhoods, where the initial screening decision is worth refining.
- $M_{\max}$**, maximum resampled target count.** After support-mass selection, we cap the number of target features sent to selective resampling at $M_{\max}$, providing a hard compute budget. In the default setting this cap is inactive; for example, FineWeb layer 34 with $(B, m) = (32, 8)$ selects 367.75 targets on average, well below $M_{\max} = 4096$. In the extreme of $(B, m) = (1, 1)$, the cap would be used, but this setting is beyond the compute constraints used in this work.

Our default pipeline uses $p = 0.99$, $B = 32$, $m = 8$, which were selected via parameter screens as effective operating point balancing graph size and computational cost. The hyperparameter variations as well as graph-overlap statistics are provided in Figure 6, Figure 5, and Table 5.

A.2. LASSO penalty calibration

We use two distinct LASSO calibration strategies depending on dataset. By default, we recommend the use of target-degree-based calibration (discussed below), when data is sufficiently large.

After support-aware screening fixes the candidate neighborhood $\widehat{C}_j$ for each target feature j, we calibrate the LASSO penalty to control sparsity in the directed proposal graph. For broad-corpus runs such as FineWeb, we use a single global penalty λ per layer/run. We initialize the search at a standard high-dimensional scaling

$$\lambda_0 = c\sqrt{\frac{\log d}{n}},$$

where d is the number of SAE features, n is the number of regression rows, and c is a fixed scale constant. We then perform a geometric binary search over a multiplicative range around λ_0. At each candidate λ, we fit nodewise LASSO on a sampled calibration subset of target nodes, candidate features, and rows, and measure the mean number of nonzero coefficients per target. For FineWeb, we select the target degree $d_0 = 10$ by default because we observe relative insensitivity to this parameter under the standard hyperparameter set (Table 4).

For smaller and more domain-constrained datasets such as WMDP-bio, target supports and candidate-union row counts are more heterogeneous. In this setting, we instead use a support-scaled penalty

$$\lambda_j = c\sqrt{\frac{\log |\widehat{C}_j|}{n_j^{\text{eff}}}},$$

where $|\widehat{C}_j|$ is the retained candidate count for target j, and n_j^{eff} is the number of support-aware regression rows available for that target. This keeps the penalty sensitive to the local amount of evidence available for each target rather than forcing a single global average-degree calibration across targets with very different support.

A.3. Semantic coherence of graphs

To better understand semantic coherence between features, we quantified module semantic coherence using Neuronpedia feature descriptions. For each dataset and layer, we ran Louvain community detection on the final stable graph, fetched Neuronpedia descriptions for all module nodes when available, and embedded the descriptions using MiniLM-L6-v2 (sentence-transformers, 2024) sentence encoder. We then computed the pair-weighted mean cosine similarity between label embeddings for all feature pairs within each module and compared it to a sampled cross-module baseline. Statistical significance was assessed with a module-preserving label-shuffle null. Module assignments and sizes were held fixed, while labels were randomly permuted across nodes. We used 1000 permutations and report both raw permutation (p)-values and Bonferroni-corrected values across the six dataset-layer tests.

Across five of six dataset-layer settings (Table 3), graph modules are systematically more semantically coherent by cosine similarity than expected under random label assignment. The five positive cases remain significant after Bonferroni correction across all six tests. FineWeb layer 22 is the exception, showing no detectable semantic enrichment.

A.4. Synthetic oracle validation

To evaluate whether the graph-estimation procedure can recover known compositional structure under controlled conditions, we additionally constructed a synthetic oracle setting in which the relevant latent variables were directly available or provided through a fixed oracle representation.

The synthetic latent data included atomic features, duplicate features, and composite features generated through parent-feature interactions. This setup was intended to isolate the downstream graph-recovery procedure from the separate problem of learning an SAE representation.

In the direct latent condition, our pipeline was applied directly to the ground-truth latent activations. In this setting, we recovered the planted compositional motifs cleanly, with motif recall, precision, and F1 all equal to 1.0. Edge recall was also 1.0, indicating complete recovery of the ground-truth parent-to-composite structure, although edge F1 was lower (≈ 0.545) due to additional predicted edges outside the planted motif structure.

We additionally evaluated a fixed oracle-representation control in which the graph pipeline operated directly on a synthetic hidden representation rather than on learned SAE activations. This condition also recovered strong compositional structure, with motif recall and motif F1 of approximately 0.833, edge recall of 0.917, and edge F1 of approximately 0.60. Under the same evaluation, a cosine-similarity baseline failed to recover the planted motifs.

We intentionally omit SAE-mediated oracle-representation experiments from this appendix because they primarily probe SAE training quality rather than the downstream graph estimator itself. In preliminary experiments, failures in these settings were dominated by imperfect factorization, feature splitting, and low feature purity within the learned SAE representation.

A.5. Compute Requirements

All development for this work was conducted on a Nvidia GB10 node with 128Gb of VRAM. In the reported setting of gemma2:27b, a Nvidia GPU with 64Gb of RAM should be suitable for full replication of results at int8 model quantization, with the highest memory requirement existing at the initial stage of activation sampling. An individual trial of 10-fold repeats with 3 null repeats is expected to take a maximum of 45 minutes wall clock (35-40 minutes average) on this node configuration. Storage use varies, but is below 50Gb disk space for any 10-fold repeat with 3 null repeats tested in this work. This is the maximum expected for primary graph generation on any dataset. Smaller experiments, like the similarity matched baselines, will also use no more than this amount of compute. We estimate that a full replication of all experiments included in this work will take 0.5-1 GPU-day on an equivalent GB10 node, producing no more than 750Gb of intermediate data, although this is more likely to be in the range of 300-600Gb. The full paper used substantially more compute over the course of development, generating 10Tb+ of intermediate files and using 1-2 GPU-months of compute, as our original design did not contain the compute optimizations discussed in the final work here.

A.6. Existing Assets and Licenses

This work uses existing model, SAE, dataset, and interpretability-label assets. We used the Gemma 2 27B base model (Gemma Team, 2024), distributed by Google under the Gemma Terms of Use. We used Gemma Scope

Dataset	Layer	Modules	Neuronpedia labels	Within	Across	Δ	z	p	$p_{\mathrm{Bonf.}}$
FineWeb	10	14	19,879/19,900 (99.9%)	0.1255	0.1182	0.0073	6.69	0.000999	0.00599
FineWeb	22	20	7,209/7,257 (99.3%)	0.1250	0.1254	-0.0004	-0.22	0.583	1.000
FineWeb	34	27	9,694/9,698 (100.0%)	0.1150	0.1050	0.0100	8.81	0.000999	0.00599
WMDP-bio	10	31	7,455/7,457 (100.0%)	0.1315	0.1185	0.0130	13.24	0.000999	0.00599
WMDP-bio	22	57	5,261/5,263 (100.0%)	0.1397	0.1264	0.0133	12.58	0.000999	0.00599
WMDP-bio	34	23	6,056/6,056 (100.0%)	0.1119	0.0995	0.0124	15.63	0.000999	0.00599

Table 3. Semantic-coherence check for Louvain modules. Neuronpedia feature descriptions were embedded with a MiniLM-style encoder. Within and Across report pair-weighted mean cosine similarity for feature-label embeddings within graph modules and for sampled feature pairs from different modules, respectively; $\Delta = \mathrm{Within} - \mathrm{Across}$. The z and p columns compare the observed within-module similarity to a module-preserving label-shuffle null with 1000 permutations. The value $p = 0.000999$ is the permutation resolution floor. $p_{\mathrm{Bonf.}}$ applies a Bonferroni correction across the six planned dataset–layer checks. Five of six settings remain significant after correction; FineWeb layer 22 is the exception.

sparse autoencoders (Lieberum et al., 2024), specifically the Gemma 2 27B pretrained residual-stream SAEs at width 131k for layers 10, 22, and 34, distributed through Hugging Face under CC-BY-4.0. We also use the sentence-transformers all-MiniLM-L6-v2 model for semantic similarity analysis, which is based on (Wang et al., 2020) and distributed through HuggingFace under the Apache 2.0 license.

For data, we used FineWeb (Penedo et al., 2024) as a broad webtext corpus; FineWeb is distributed under the Open Data Commons Attribution License (ODC-By) v1.0 and is derived from Common Crawl, whose terms of use also apply. We used the WMDP-bio split of the WMDP benchmark (Li et al., 2024); WMDP is distributed under the MIT license.

For feature-label inspection and figure annotations, we used Neuronpedia (Lin, 2023). Neuronpedia is an open-source interpretability platform with MIT-licensed source code; feature descriptions and labels were accessed from the public Neuronpedia site/API and are credited to Neuronpedia. We do not redistribute the Gemma 2 model weights, Gemma Scope SAE weights, MiniLM-L6-v2, FineWeb, WMDP, or Neuronpedia itself. Any released graph artifacts are derived analysis outputs from these assets and should be interpreted subject to the upstream licenses and terms above.

d_0	$\bar{\lambda}$	Dir. edges	Graph edges	$k^\star$	Nodes	Edges	CCs	LCC	Q	Jaccard
5	5.13e-05	353,466	182,716	3	9,705	18,025	7	0.998	0.850	0.999
10	5.13e-05	353,433	182,718	3	9,698	18,025	7	0.998	0.850	1.000
20	5.13e-05	353,408	182,696	3	9,696	18,015	7	0.998	0.848	0.998

Table 4. Sensitivity of the layer-34 FineWeb stable graph to the target average directed degree d_0 used for global LASSO penalty calibration. Jaccard overlap is measured against the default $d_0 = 10$ stable edge set.

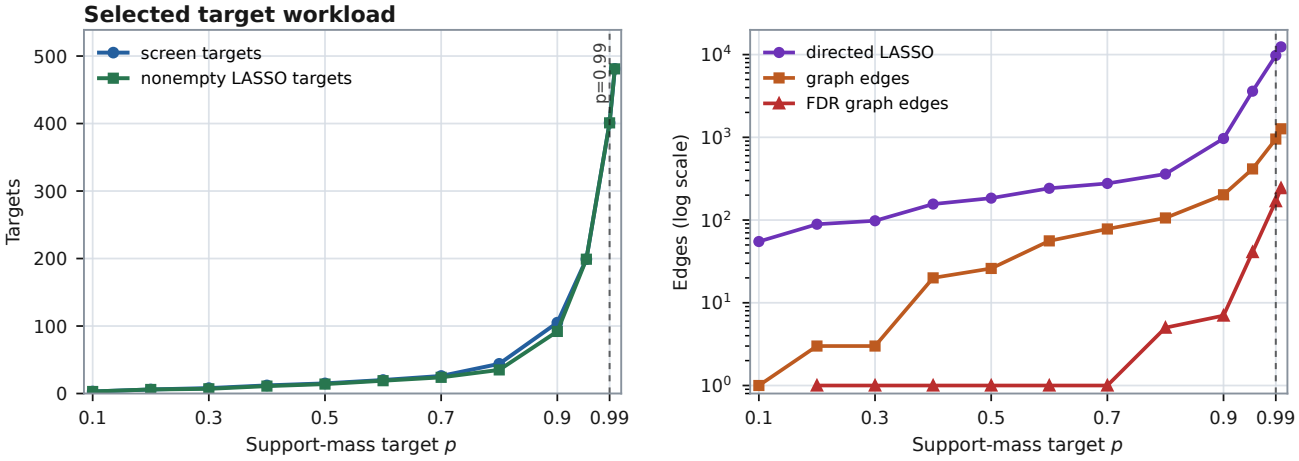

Figure 5. Support-mass sweep for the selective resampling target p. Left: increasing p increases the number of selected screening targets and nonempty LASSO targets. Right: downstream directed LASSO edges, symmetrized graph edges, and FDR-filtered graph edges grow sharply as p approaches one. The default $p = 0.99$ retains most ambiguous support mass while avoiding the extreme tail of screening growth.

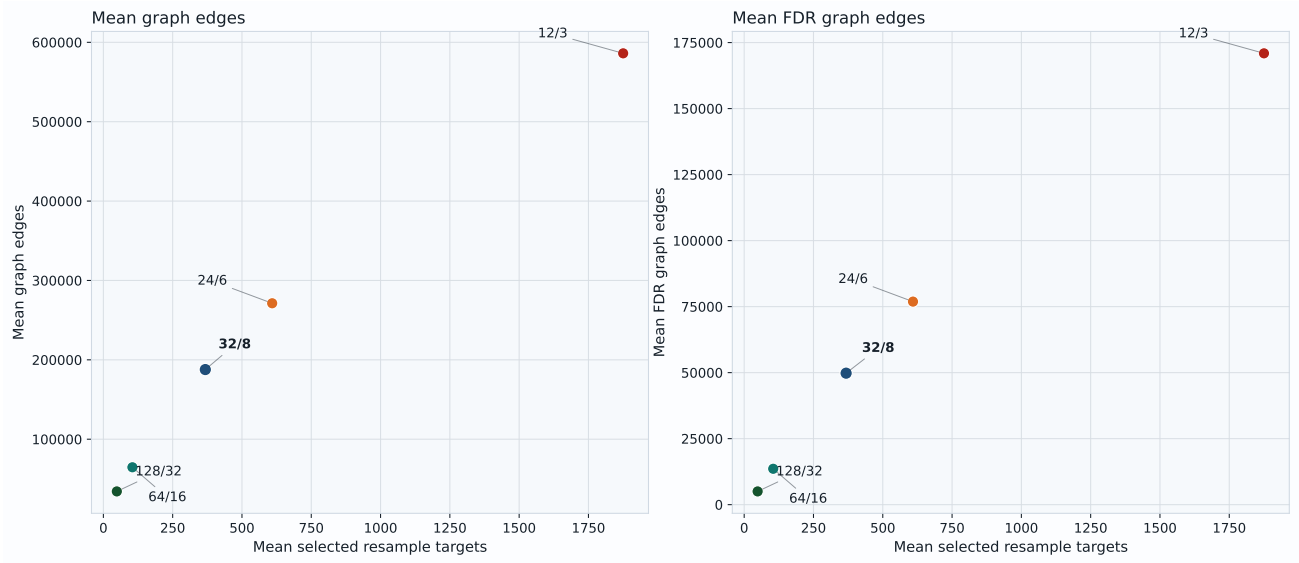

Figure 6. Selective-screening gate frontier for B, m on FineWeb layer 34. Each point varies the support-aware screening gates, where B is the minimum target support count and m is the minimum coarse-candidate count. Looser gates select many more resampling targets and produce substantially larger graphs, while stricter gates collapse the proposal set. The default $B = 32, m = 8$ lies between excessively large and near-empty proposal sets.

(B, m)	Targets	Graph	FDR	$k^\star$	Stable	Cand. pairs	Cand. Jac.	Stable Jac.
$(12, 3)$	1,875.25	586,138	170,928	3	5,963	5,308,363	0.312	0.098
$(24, 6)$	609.00	271,257	76,897	3	2,705	2,432,885	0.680	0.107
$(32, 8)$	367.75	187,736	49,796	3	18,025	1,654,906	1.000	1.000
$(64, 16)$	104.75	64,645	13,588	2	8,916	549,295	0.332	0.124
$(128, 32)$	48.50	34,229	5,015	2	4,755	276,569	0.167	0.066

Table 5. Selective-screening gate frontier for FineWeb layer 34 at fixed evidence budget. Candidate pairs are the four-fold union of directed effective LASSO candidate pairs, and candidate Jaccard is measured against the canonical $(B, m) = (32, 8)$ setting. Stable Jaccard is measured against the canonical setting stable-edge set.