CROSS-LAYER CLUSTERING FOR STOCHASTIC PARAMETER DECOMPOSITION

Saman Seshadri[‡] Algoverse
Jack Digilov[*] Algoverse
Sean Esla[*] Algoverse
Nathan Zixia Hu Stanford University
Michael Ivanitskiy Colorado School of Mines
Pablo Bernabeu Perez[‡] Algoverse

ABSTRACT

Mechanistic interpretability seeks to decompose neural networks into interpretable circuits. Stochastic parameter decomposition (Bushnaq et al., 2025, SPD) yields sparse, atomic subcomponents within layers but does not capture the multi-layer pathways driving complex behavior. We propose a cross-layer spectral clustering framework that automatically discovers these distributed mechanisms by analyzing co-activation patterns across inputs. By measuring the Pearson correlation of importance scores between subcomponents, we construct a similarity graph that links disjoint parts of the network contributing to the same computational task. On synthetic models with known circuits, our method successfully recovers the ground-truth mechanistic structure confirming its ability to identify cross-layer dependencies. When applied to small language models, we find multi-layer clusters whose top-activating examples suggest consistent linguistic functions (e.g., tracking salient entities and tense morphology). These clusters serve as high-quality hypotheses for follow-up causal tests, providing a scalable step toward discovering system-level mechanisms in language models.

1 INTRODUCTION

As large language models grow in capabilities and see widespread adoption, a central question arises: how can we decompose these complex systems into human-understandable mechanisms (Bereska & Gavves, 2024; Ferrando et al., 2024; Sharkey et al., 2025)? If we want to ensure both trustworthiness and controllability, we need methods that move beyond black-box performance to offer a transparent view of a network's constituent computations.

Current interpretability research largely focuses on activation-space analysis, identifying features within hidden states of the model (Bricken et al., 2023; Marks et al., 2024). Recently, however, parameter-space methodologies have emerged as a promising alternative, decomposing model weights directly into discrete, functional subcomponents. Stochastic parameter decomposition (Bushnaq et al., 2025, SPD) represents parameters as sums of sparse matrices, isolating granular, atomic circuits. However, existing SPD frameworks are limited to within-layer decompositions, failing to capture mechanisms that span multiple layers.

We introduce a cross-layer spectral clustering framework designed to bridge this gap by automatically discovering multi-layer mechanisms. Our approach rests on the hypothesis that subcomponents involved in the same computational pathway exhibit correlated patterns of importance as input data changes. We validate our method on synthetic toy models (Elhage et al., 2022), for which our clustering recovers expected cross-layer mechanistic structures. When applied to small language models, where underlying mechanisms are unknown, our approach identifies candidate multi-layer circuits whose top-activating examples suggest interpretable clusters.

[*] Equal contribution.
[†] Code available at https://github.com/goodfire-ai/spd/tree/clustering/algoverse/tms-spectral-v2/spd/clustering
[‡] Senior author.

These results suggest that parameter-space interpretability can be extended across layers, yielding modular explanations for non-trivial neural behaviors. By utilizing efficient correlation-based metrics, we provide a scalable alternative to computationally expensive causal search methods. Our framework could serve as a *hypothesis generator* that filters the search space into plausible multi-layer circuits for targeted intervention experiments. This transition from analyzing isolated layers to identifying distributed circuits represents a step toward understanding frontier language models.

Contributions

- **Cross-layer clustering framework:** We introduce a spectral clustering method based on SPD component co-activation to automatically identify distributed mechanisms across layers.
- **Synthetic validation:** We test our framework on toy models where it accurately recovers ground-truth cross-layer circuits.
- **LM application:** Our method identifies interpretable multi-layer clusters in small language models linked to specific linguistic functions.

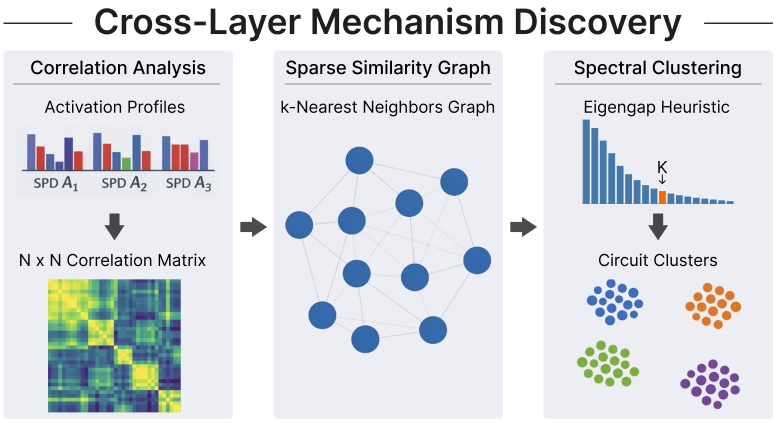


Figure 1: Cross-layer circuit discovery pipeline: starting from SPD subcomponents' activation profiles, we compute a cross-layer correlation matrix, build a sparse k-nearest-neighbors similarity graph, and apply spectral clustering with an eigengap heuristic to recover multi-layer circuit clusters that correspond to distributed circuits in the network.

2 Background

Stochastic parameter decomposition (Bushnaq et al., 2025, SPD) is a parameter-space interpretability technique that aims to decompose model weights into atomic subcomponents. It represents each layer's weight matrix W^l as a sum of rank-one factors, each associated with a learned importance predictor $g_c^l(x)$ that determines how active the subcomponent (l, c) should be for a given input x. These importance predictors are implemented as small MLP gates that take pre-activation signals as input and output values in $[0, 1]$ via a hard sigmoid.

Earlier gradient-based approaches, such as attribution-based parameter decomposition (Braun et al., 2025, APD), decompose parameters using gradient-derived attributions combined with a batch top-k selection rule. While effective for simple networks, APD suffers from mechanism mixing, parameter shrinkage, and significant computational overhead due to its reliance on gradients. SPD addresses these limitations through its stochastic masking mechanism, removing the need for a top-k hyperparameter, avoiding parameter shrinkage, and scaling effectively to larger models.

Spectral clustering (von Luxburg, 2007) is a graph-based technique that excels at detecting well-separated communities in a similarity graph, making it suitable for uncovering mechanisms with strong internal co-activation and weak external coupling. It constructs a similarity graph, computes its normalized Laplacian, and embeds the data using the leading eigenvectors, on which standard algorithms like k-means can then identify clusters.

Given a similarity matrix S with affinities s_{ij} between nodes i and j, the normalized graph Laplacian is $\mathcal{L} = I - D^{-1/2}WD^{-1/2}$, where D is the diagonal degree matrix with entries $D_{ii} = \sum_j w_{ij}$. The number of clusters is often determined using the eigengap heuristic, which selects the largest gap between consecutive eigenvalues λ_k and λ_{k+1} in the generalized eigenproblem $Lv = \lambda Dv$.

3 Problem Setting

Let $f : \mathcal{X} \to \mathcal{Y}$ be a trained neural network with L layers and weight matrices $W^l \in \mathbb{R}^{d_{l+1} \times d_l}$ for $l = 1, \ldots, L$. We assume the network has been decomposed using SPD, yielding a rank-one factorization of each weight matrix $W^l \approx \sum_{c=1}^{C_l} U_c^l V_c^{l\top}$, where $U_c^l \in \mathbb{R}^{d_{l+1}}$ and $V_c^l \in \mathbb{R}^{d_l}$ are the left and right factors of subcomponent (l, c), and C_l is the number of subcomponents in layer l.

Each subcomponent is paired with a learned importance predictor $g_c^l : \mathcal{X} \to [0, 1]$, parameterized as $g_c^l(x) = \sigma_G\big(\gamma_c^l(h_c^l(x))\big)$, $h_c^l(x) = \sum_j V_{c,j}^l a_j^l(x)$, where $a^l(x)$ is the input activation vector to layer l, γ_c^l is a learned gating network, and $\sigma_G : \mathbb{R} \to [0, 1]$ is a bounding function (e.g., hard sigmoid, soft sigmoid, or sparsemax). The inner activation $h_c^l(x)$ captures how strongly subcomponent (l, c) is engaged by input x. For notational convenience, we index all subcomponents globally by $i \in \{1, \ldots, N\}$, where $N = \sum_{l=1}^{L} C_l$ and each i corresponds to a unique pair (l_i, c_i). We write $g_i(x) := g_{c_i}^{l_i}(x)$ for the importance predictor of global subcomponent i.

Cross-Layer Mechanisms A *cross-layer mechanism* is a subset $\mathcal{M} \subseteq \{1, \ldots, N\}$ of subcomponents spanning multiple layers that collectively implement a coherent computational function. Our central hypothesis is:

Hypothesis: Co-activation indicates shared mechanism membership

Subcomponents belonging to the same computational pathway exhibit correlated importance scores across an input distribution. Formally, if $i, j \in \mathcal{M}$ for some mechanism $\mathcal{M}$, then the correlation $\rho_{ij} := \mathrm{corr}\big(g_i(X), g_j(X)\big)$ should be significantly positive for a representative dataset $X = \{x_m\}_{m=1}^M$.

Our hypothesis extends mediator-based interpretability ideas (Vig et al., 2020; Mueller et al., 2025) from activation space to parameter space, exploiting SPD's learned importance functions to expose cross-layer functional groupings.

4 Method

We discover cross-layer mechanisms by clustering SPD subcomponents according to their co-activation patterns across inputs. The method consists of three steps: (1) computing correlation-based similarity between all subcomponent pairs, (2) constructing a sparse k-nearest-neighbor graph from these correlations, and (3) performing spectral clustering to recover cross-layer mechanisms.

Activation Profiles and Correlation-Based Similarity. For each subcomponent i and dataset $X = \{x_m\}_{m=1}^M$, we define its *activation profile* as $A_i := [g_i(x_1), g_i(x_2), \ldots, g_i(x_M)] \in \mathbb{R}^M$, which captures how the subcomponent's importance varies across inputs. The similarity between subcomponents i and j is given by the Pearson correlation of their profiles, $\rho_{ij} := \mathrm{corr}(A_i, A_j)$. Because Pearson correlation is scale-invariant and robust to magnitude differences, it serves as an appropriate measure of shared activation patterns, consistent with our hypothesis (see Section 3). In practice, we estimate ρ_{ij} using the sample correlation computed over the M datapoints, obtaining the full $N \times N$ correlation matrix in a single vectorized operation. For N in the range of tens to hundreds (typical of SPD decompositions in smaller models), this procedure remains computationally tractable.

Sparse Similarity Graph Construction. Given the correlation matrix $\{\rho_{ij}\}$, we construct a sparse k-nearest-neighbor similarity graph for spectral clustering. We convert raw Pearson correlations ρ_{ij} to non-negative affinities by clamping negative values to zero: $s_{ij} = \max(0, \rho_{ij})$. This reflects our

assumption (Section 3) that mechanisms are characterized by positive co-activation; anti-correlated components (e.g., switch-like subcircuits where one activates when another deactivates) are treated as unconnected rather than grouped together. We leave exploration of signed-graph spectral clustering methods to future work. For each node i, we retain only edges to its k nearest neighbors (highest s_{ij} values), setting other weights to zero. We then symmetrize using mutual k-NN: we keep edge (i, j) only if both i is among j's k neighbors *and* j is among i's k neighbors. This ensures the graph is undirected and reduces the edge count from $O(N^2)$ to $O(kN)$, making subsequent spectral decomposition tractable.

Spectral Decomposition and Clustering. We compute the degree matrix D with $D_{ii} = \sum_j s_{ij}$ and solve the generalized eigenproblem $Lv = \lambda Dv$, where $L = D - W$ is the unnormalized graph Laplacian. This is equivalent to using the normalized Laplacian $\mathcal{L} = D^{-1/2}LD^{-1/2}$ after appropriate rescaling. We use the *eigengap heuristic* to automatically select the number of clusters K: we choose $K = \arg\max_k(\lambda_{k+1} - \lambda_k)$, identifying the largest gap in the eigenvalue spectrum. We then form an embedding matrix $E \in \mathbb{R}^{N\times K}$ using the first K eigenvectors as columns and apply k-means to the rows of E to obtain K clusters. This provides an initial partitioning; for interpretability studies, we manually inspect cluster semantics (via top-activating examples, Section 5) and may adjust hyperparameters based on coherence. When the k-NN graph has multiple connected components, we perform spectral clustering on the largest connected component and assign remaining nodes to their nearest cluster in the correlation space post-hoc.

5 Experimental Setup

We validate our clustering framework on models ranging from synthetic toy systems with known structure to small language models where mechanisms must be discovered.

Toy Model of Superposition with Hidden Identity ($\text{TMS}_{\text{5-2+ID}}$). To test whether our clustering method can identify compositional structure, we study a toy model introduced by Bushnaq et al. (2025) where an identity transformation is explicitly inserted between the encoder and decoder $\hat{x} = \text{ReLU}(W^\top IWx + b)$ where $W \in \mathbb{R}^{m_1\times m_2}$ with $m_1 = 2$ hidden dimensions and $m_2 = 5$ input features, and $I \in \mathbb{R}^{m_1\times m_1}$ is an identity matrix which displays superposition (a phenomenon caused when there are more features than dimensions forcing individual neurons to be polysemantic). The model tests for feature splitting, a phenomenon in sparse dictionary learning where features depend on dictionary size (Bricken et al., 2023). The expected ground-truth decomposition comprises $m_2 + m_1 = 7$ subcomponents: $m_2 = 5$ subcomponents corresponding to the columns of W (each causally important for a single input feature), and $m_1 = 2$ subcomponents for the identity matrix I (which should have causal importance for nearly all inputs).

Toy Model of Superposition ($\text{TMS}_{\text{40-10}}$). Following Elhage et al. (2022), we evaluate on an autoencoder-style model that compresses sparse input vectors into lower-dimensional hidden representations $\hat{x} = \text{ReLU}(W^\top Wx + b)$ where $W \in \mathbb{R}^{m_1\times m_2}$ with $m_1 = 10$ hidden dimensions and $m_2 = 40$ input features. The model is trained to reconstruct sparse inputs sampled from one-hot m_2-dimensional feature vectors with activations scaled uniformly over $[0, 1]$. Since $m_1 < m_2$, the model must compress representations through a bottleneck, learning to represent features in superposition when trained on sufficiently sparse data distributions. Based on Bushnaq et al. (2025), the ground-truth mechanisms in this model correspond to $m_2 = 40$ rank-one matrices, where each mechanism isolates a single column of W that is causally important if and only if the corresponding input feature is active.

Two-Layer Residual MLP. To study mechanisms distributed across multiple layers, we evaluate on a residual MLP architecture introduced by Braun et al. (2025) and further studied in Bushnaq et al. (2025). The model is trained to approximate $y_i = x_i + \text{ReLU}(x_i)$ for $n = 100$ sparsely activating input features, where each $x_i \in [-1, 1]$. The architecture consists of two residual MLP blocks with a shared residual stream $\hat{y} = W_E^\top \left(W_{\text{out}}^{(2)}h_2 + W_{\text{out}}^{(1)}h_1 + W_E x\right)$ where $h_1 = \text{ReLU}(W_{\text{in}}^{(1)}W_E x)$ and $h_2 = \text{ReLU}(W_{\text{in}}^{(2)}(W_{\text{out}}^{(1)}h_1 + W_E x))$. Here, $W_E \in \mathbb{R}^{d_{\text{embed}}\times n}$ is a fixed random embedding matrix with unit-norm rows, and $W_{\text{in}}^{(\ell)} \in \mathbb{R}^{m_\ell\times d_{\text{embed}}}$, $W_{\text{out}}^{(\ell)} \in \mathbb{R}^{d_{\text{embed}}\times m_\ell}$ are trainable weights

with $d_{\text{embed}} = 1000$ and $m_1 = m_2 = 25$ neurons (i.e., 50 neurons across both layers). Crucially, the task requires computing more ReLU functions ($n = 100$) than available neurons ($m = 50$), forcing the model to use multiple neurons polysemantically to compute each input-output mapping.

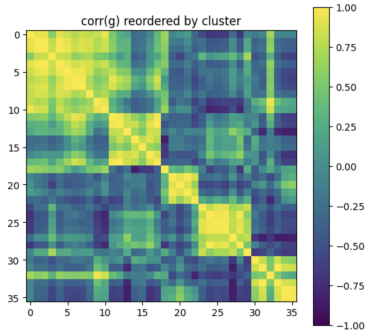
Three-Layer Residual MLP. We also evaluate on a deeper MLP variant that spreads $m = 51$ neurons over three MLP blocks to compute $n = 102$ input-output functions (Bushnaq et al., 2025). The architecture follows the same residual accumulation pattern, with each layer $\ell \in \{1, 2, 3\}$ contributing $m_\ell = 17$ neurons. Each MLP block has input weights $W_{\text{in}}^{(\ell)} \in \mathbb{R}^{17\times1000}$ and output weights $W_{\text{out}}^{(\ell)} \in \mathbb{R}^{1000\times17}$. In both the two- and three-layer models, the ground-truth mechanisms comprise: (1) n subcomponents in the MLP input matrices $W_{\text{in}}^{(\ell)}$, each corresponding to one input feature and distributed across all layers; and (2) a single high-rank component spanning all $W_{\text{out}}^{(\ell)}$ matrices, which projects MLP activations back to the residual stream and should co-activate for all inputs. This model is already challenging: Braun et al. (2025) report that APD struggled to decompose models with more than two layers due to hyperparameter sensitivity.

SimpleStories. Finally, we apply our method to SimpleStories (Finke et al., 2025), a 4-layer, 1.25M parameter, decoder-only transformer designed as a "model organism" for mechanistic interpretability research. The model adopts a minimal LLaMA-like architecture (d_{model} = 128, n_{heads} = 4) and a custom WordPiece tokenizer with a $4,096$-word vocabulary optimized for the SimpleStories corpus, a dataset comprising ~2M synthetic narratives with controlled variation in topic, style, and structure.

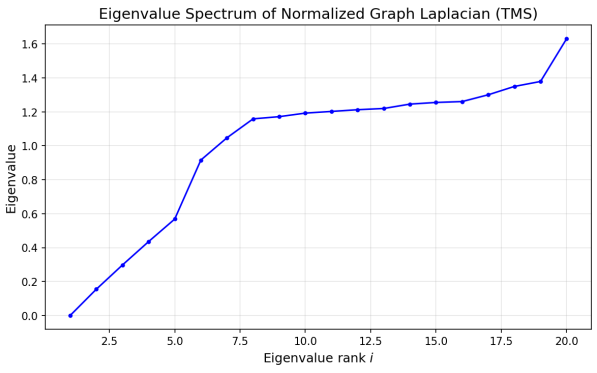
6 RESULTS

We present decomposition quality and clustering results across our model suite, progressing from simple toy models to realistic language models.

$\text{TMS}_{\text{5-2+ID}}$. We first apply our clustering method on $\text{TMS}_{\text{5-2+ID}}$, which inserts an explicit identity matrix $I \in \mathbb{R}^{2\times2}$ between encoder-decoder: $\hat{x} = \text{ReLU}(W^\top IWx + b)$ ($W \in \mathbb{R}^{2\times5}$). We expect 5 singleton feature clusters plus 1 identity cluster of 2 components. Of the SPD subcomponents in this toy model, 36 remain active if their maximum gate value exceeds 0.1 (thresholding discussed in the Appendix D). Spectral clustering yields exactly 5 cross-layer clusters (Figure 2a), with block-diagonal correlation structure and eigenvalues $\lambda_1 \ldots \lambda_5$ clearly separated from the bulk spectrum (Figure 2b). However, the identity functionality is distributed across all 5 clusters rather than isolated in a single group (expected: 5 feature + 1 identity cluster of 2), indicating entanglement with feature pathways.



(a) Correlation heatmap ordered by cluster. Diagonal blocks show strong within-cluster co-activation.



(b) Eigenvalue spectrum ($N = 36$, $k = 10$). First 5 eigenvalues separate clearly from bulk.

Figure 2: $\text{TMS}_{\text{5-2+ID}}$ clustering results for 36 active components. (a) Correlation structure reveals 5 distinct clusters with minimal cross-cluster correlation. (b) Spectral gap confirms 5-cluster solution.

$\text{TMS}_{\text{40-10}}$. We also evaluate our method on the $\text{TMS}_{\text{40-10}}$, which compresses 40 sparse input features into a 10-dimensional bottleneck via $\hat{x} = \text{ReLU}(W^\top Wx + b)$, forcing superposition. From

the SPD decomposition, we expect 40 feature-computing mechanisms, each pairing one encoder and one decoder subcomponent. From 394 SPD subcomponents (197 per matrix), filtering yields 80 active components (max gate > 0.1). These form exactly 40 encoder-decoder pairs via spectral clustering, with each pair showing 100% feature agreement on structured validation blocks (samples $200k$ through $200(k+1)-1$ for feature k).

Two-Layer Residual MLP. We apply our clustering framework to the two-layer residual MLP. From the SPD decomposition, we expect multiple feature-computing mechanisms in the input matrices and a single identity-projection component spanning the outputs and linking MLP activations back to the residual stream. Out of 1600 SPD-derived subcomponents, 251 are active (max gate > 0.2), with 201 in mlp_in layers and 50 in mlp_out layers. Clustering yields 101 clusters across 98 connected components, recovering 97 of 100 ground-truth feature mechanisms as size-2 clusters, each containing one mlp_in subcomponent from each layer, with 100% type purity. Cross-layer subcomponents show perfect Pearson correlations ($\rho = 1.0$) and cosine similarity of 1.0 between top-activating inputs, confirming they implement identical feature computations. We also recover the size-50 identity-projection mechanism (all `mlp_out` subcomponents) with complete type purity, balanced across layers (25 each), and dense co-activation across all inputs. Representative correlation and cosine validation results are shown in Appendix G.

Three-Layer Residual MLP. We evaluate our method on the three-layer residual MLP, which distributes 51 neurons across 102 input-output functions. From the decomposition, we expect to find layer-spanning feature-computing mechanisms in the input matrices and a single identity-projection component across the output matrices linking activations to the residual stream. Spectral clustering on the 3000 SPD subcomponents (379 active with max gate > 0.1) uncovers over 80 cross-layer triplets (one `mlp_in` subcomponent per layer), with over 65 exhibiting perfect cosine alignment ($|\cos| = 1.0$) and over 75 showing near-perfect alignment ($|\cos| \geq 0.99$). All qualifying clusters show 100% argmax-input agreement, confirming identical feature computations. Graph coherence is strong: 99.8% of k-NN edges stay within clusters ($\bar{\rho}_{\text{within}} = 0.175 \gg \bar{\rho}_{\text{between}} = 0.024$). The 17 alive `mlp_out` subcomponents per layer form a single rank-17 component with dense co-activation across inputs, matching the expected high-rank identity mechanism.

Cluster ID	Top Activating Tokens (Mean Activation)	Preliminary Interpretation
547	`wanted` (5.72), `saw` (5.72), `said` (5.65), `felt` (5.57), `found` (5.50), `would` (4.84), `were` (4.74), `had` (4.49)	Uniformly activates on past-tense action verbs common in storytelling contexts.
595	`he` (5.58), `mia` (5.57), `she` (5.54), `alex` (5.47), `i` (5.40), `boy` (5.36), `they` (5.36), `girl` (4.94)	Tracks narrative agents via both proper nouns and pronominal references.
7	`girl` (3.98), `alex` (3.98), `boy` (3.92), `with` (3.63), `felt` (3.44), `named` (3.07), `mia` (2.92), `said` (2.73), `heart` (2.24)	Activates on character introductions and associated emotional states.
4	`like` (2.13), `one` (1.32), `up` (1.14), `into` (1.02), `said` (0.76), `filled` (0.74), `joy` (0.58), `together` (0.48)	Handles comparative language and positive emotional descriptors.
805	`back` (1.74), `up` (1.03), `together` (0.96), `-ed` (0.89), `them` (0.85), `like` (0.77), `him` (0.75), `joy` (0.62), `heart` (0.58)	Processes scenes of collective action, movement, and emotional resolution.

Table 1: Top-activating tokens for five clusters. Each cluster represents a learned internal feature. Token entries show mean activation strength. Higher activation indicates the token strongly triggers that cluster's pattern.

SimpleStories. SPD decomposes 28 modules across the 4 transformer layers (7 projections per layer: q, k, v, o for self-attention, plus gate, up, down for the MLP). With C = 2000 subcomponents per module, this yields 56,000 total components. After applying dead component filtering with a maximum gate threshold of 0.1, we retain 8,425 active components (15.0% survival rate). In particular, we find that MLP projections retain more active components than attention projections, consistent

with the hypothesis that feed-forward layers implement more diverse feature transformations (see Table 7 in Appendix I to see the distribution of active components across module types). Additionally, we report dead component progression over SPD training for layers 0 and 3 in Figures 4a and 4b.

From these active components, we construct a mutual k-nearest neighbor graph with $k = 4$ using ReLU-clamped correlation affinities. The resulting graph contains $13,408$ edges with $3,041$ isolated nodes. Connected component analysis reveals $4,063$ connected components, with the largest containing 129 nodes. Spectral clustering within each connected component (using the eigengap heuristic with $K_{\max} = 12$) produces $4,784$ total clusters. Cluster size distribution follows a heavy-tailed pattern: the largest clusters contain 22, 18, 16, 15, and 14 components respectively, while $3,041$ clusters are singletons.

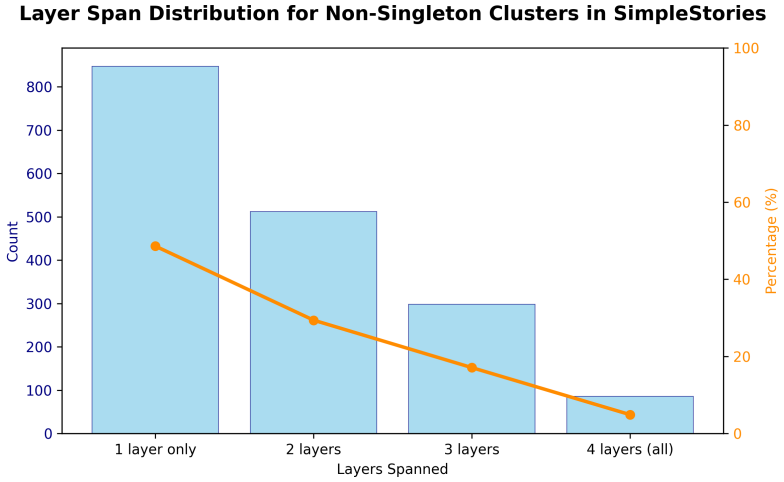
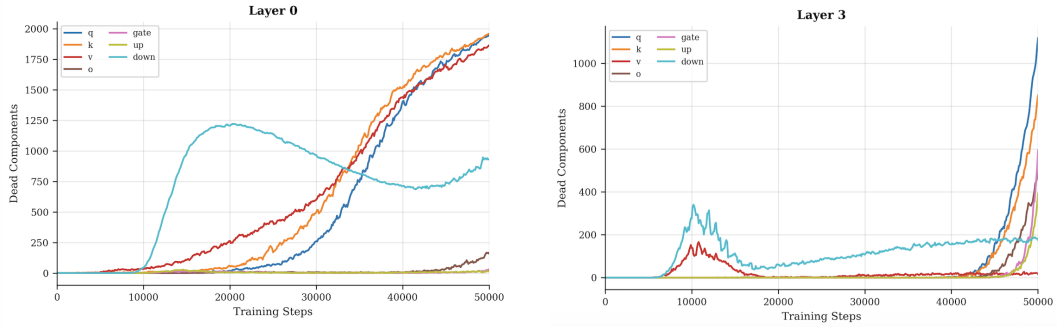


Figure 3: Layer span distribution for non-singleton clusters in SimpleStories. Bars show absolute counts; orange line shows percentages.

In Figure 3 we characterize the layer coverage of clusters that span multiple layers and module types. In particular, we highlight that over half (51.4%) of non-singleton clusters span two or more layers, with 86 clusters (4.9%) spanning all four transformer layers, suggesting that the clustering captures distributed mechanisms rather than layer-local computations.



(a) Layer 0

(b) Layer 3

Figure 4: Dead component distribution across modules in SimpleStories for layers 0 and 3.

We further analyze the composition of clusters by module type and find that there are 412 MLP-pure clusters (containing only MLP components), 89 attention-pure clusters (containing only attention components), and 242 mixed clusters (both MLP and attention components). While the semantic coherence of these clusters provides strong qualitative evidence for mechanism discovery, **we emphasize that these interpretations are preliminary**. The "max-activating" heuristic relies on correlation, which, while informative, cannot definitively rule out polysemanticity or incidental co-activation

without more rigorous intervention. Validating these mechanisms as atomic computational units may require training Sparse Autoencoders (SAEs) on the residual stream to disentangle superposed features or applying causal mediation analysis to test whether ablating a cluster selectively impairs the targeted capability (e.g., entity tracking) without broader degradation. Nonetheless, the coherent groupings derived from parameter-space correlations suggest that SPD with spectral clustering offers a viable alternative or precursor to SAE-based methods for identifying macroscopic circuit structures in language models.

7 DISCUSSION

We introduce a correlation-based spectral clustering framework that extends stochastic parameter decomposition from within-layer subcomponents to cross-layer mechanisms. Our experiments demonstrate the viability of this approach across models of varying complexity, while also revealing important limitations and trade-offs.

On toy models with clean decompositions, our clustering method successfully recovers expected cross-layer structure. The eigengap heuristic automatically selects the correct number of clusters without manual tuning, and the discovered clusters align with ground-truth mechanisms. These results validate our core hypothesis that subcomponents participating in the same computational pathway exhibit correlated importance patterns across inputs, providing a sound foundation for extending the method to more complex models.

Scaling to SimpleStories, a realistic 4-layer transformer, spectral clustering produces $4,784$ clusters from over $8,000$ active components, with many clusters exhibiting coherent token activation patterns that suggest interpretable linguistic functions. Over half of the non-singleton clusters span multiple layers, and mixed MLP-attention clusters point to cross-module mechanisms that would be missed by layer-local analysis. However, as model complexity increases, cluster quality degrades due to noisier importance profiles and increased mechanism entanglement. Despite these challenges, interpretable structure emerges even in this partially sparse regime, suggesting that correlation-based clustering can serve as a useful hypothesis generator for identifying candidate circuits in realistic language models.

Across all models, we observe a fundamental fidelity-sparsity trade-off controlled by SPD's regularization strength, suggesting the need for careful calibration. Aggressive sparsity penalties yield cleaner, more interpretable clusters by forcing the decomposition to isolate distinct computational pathways, but this comes at the cost of higher reconstruction error as the model sacrifices fidelity to achieve separation. Conversely, weak sparsity preserves task performance but produces noisy, mixed clusters where multiple mechanisms remain entangled.

8 LIMITATIONS AND FUTURE WORK

Correlation-based formulation. Correlated importance does not establish *causal* membership in the same computational pathway. Two components may correlate because they respond to the same input features (e.g., both activate on long sequences) rather than because they participate in the same mechanism. Correlation captures co-occurrence but not causal dependence. Future work could validate clusters via intervention experiments (i.e., ablating entire clusters versus random component subsets) to distinguish correlational grouping from causal linkage. Despite this limitation, correlation provides a tractable and interpretable starting point for mechanism discovery.

Assumptions about SPD decomposition and hyperparameter sensitivity. Our method assumes that SPD provides faithful decompositions where subcomponents correspond to interpretable computational units. The quality of these decompositions depends critically on several hyperparameters: the strength of the importance-minimality term, the architecture of the importance predictors, and the k used in the k-NN graph. If the sparsity penalty is too weak, many subcomponents remain active and clusters become large and mixed; if it is too strong, reconstruction degrades and the learned mechanisms no longer faithfully approximate the base model. When SPD's decomposition quality degrades due to insufficient training or poor hyperparameter selection, clustering performance will also degrade. Future work could develop adaptive or data-driven procedures for setting hyperparameters or joint optimization methods that simultaneously learn decompositions and cluster assignments

Scalability to large models. Though our algorithm has near-linear complexity, applying it to very large language models with billions of parameters remains challenging. Future work could explore hierarchical clustering approaches or online algorithms that incrementally cluster components.

Validation without ground truth and cluster metrics. For language models, evaluation relies primarily on reconstruction losses, sparsity measures (e.g., L0 counts), and qualitative inspection of cluster interpretability. Developing robust quantitative metrics for circuit quality and systematically incorporating them into training and evaluation pipelines remains an open challenge.

Beyond rank-one components. Our current approach is restricted to clustering SPD's rank-one components, which may be insufficient for mechanisms that require higher-rank representations or more complex structures. Extending our framework to handle richer decomposition schemes (Braun et al., 2025) remains an important challenge for future work.

9 Related Work

Superposition and Activation-Space Features. A central challenge in mechanistic interpretability is understanding how neural networks represent more features than they have dimensions. Elhage et al. (2022) demonstrate this phenomenon of *superposition* in single-layer ReLU networks trained on sparse synthetic data, showing that networks learn to encode features in overlapping directions when inputs are sufficiently sparse. To address the resulting polysemanticity, Bricken et al. (2023) train sparse autoencoders (SAEs) on transformer activations, extracting approximately monosemantic features that correspond to interpretable concepts. While these methods provide valuable insights into feature representation, they operate in activation space and analyze each layer independently, leaving open the question of how features are implemented in the model's parameters and how they connect across layers. Beyond one-dimensional features, Engels et al. (2024) show that some representations are inherently multi-dimensional, discovering circular subspaces encoding days of the week and months. Other work explores replacing dense layers with learned sparse sublayers: Anthropic (2025) introduce sparse mixtures of linear transforms (MOLTs) that learn sparsely active transformations bridging representations between layers, while Oldfield et al. (2025) propose Mixture of Decoders (MxDs) that decompose MLPs into thousands of specialized full-rank sublayers.

Circuit Discovery. A complementary line of work focuses on discovering circuits, subgraphs of model components that implement specific behaviors. Wang et al. (2023) use causal interventions to identify the circuit responsible for indirect object identification in GPT-2 small, treating attention heads as the fundamental units of analysis. Conmy et al. (2023) automate this process with algorithms that systematically apply activation patching to discover circuits without manual effort. Marks et al. (2024) extend this approach by combining SAEs with circuit discovery, treating learned features as nodes in causal graphs to trace information flow. Lindsey et al. (2025) apply attribution graphs to Claude 3.5 Haiku, revealing mechanisms underlying multi-step reasoning, multilingual processing, and safety behaviors. These methods yield detailed accounts of model behavior but rely on activation-space analysis and often require task-specific manual effort, limiting their scalability to arbitrary mechanisms.

Parameter-Space Interpretability. Recent work has begun bridging activation-space and parameter-space interpretability. Ameisen et al. (2025) introduce cross-layer transcoders that learn features with multi-layer decoders and produce attribution graphs tracing causal paths between features; however, interpreting these graphs requires manually grouping related features into "supernodes". In parameter space, Braun et al. (2025) propose attribution-based parameter decomposition (APD), which uses gradient-derived attributions to decompose weights into functional subcomponents. Chrisman et al. (2025) introduce Local Loss Landscape Decomposition (L3D), which identifies low-rank subnetworks by decomposing the gradient of the loss in parameter space. While effective for simple networks, these gradient-based methods can suffer from mechanism mixing and computational overhead. Stochastic parameter decomposition Bushnaq et al. (2025) addresses these limitations through a stochastic masking mechanism but remains restricted to within-layer analysis.

10 CONCLUSION

We introduce a correlation-based spectral clustering framework that builds on SPD to automatically group rank-one, layer-local subcomponents into candidate multi-layer circuits. On small MLPs and toy superposition models, our clustering recovers meaningful cross-layer structure from the sparse high-fidelity decompositions SPD yields. On small transformer-style models, SPD decompositions are only partially sparse, but our method still produces clusters that give a useful starting point for mechanistic analysis.

REFERENCES

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Tristan Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelsey Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Anthropic. Sparse mixtures of linear transforms. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/bulk-update/index.html.

Leonard F. Bereska and Efstratios Gavves. Mechanistic interpretability for AI safety—a review. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ePUVetPKu6. Survey Certification, Expert Certification.

Dan Braun, Lucius Bushnaq, Stefan Heimersheim, Jake Mendel, and Lee Sharkey. Interpretability in parameter space: Minimizing mechanistic description length with attribution-based parameter decomposition. *arXiv preprint arXiv:2501.14926*, 2025.

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

Lucius Bushnaq, Dan Braun, and Lee Sharkey. Stochastic parameter decomposition. *arXiv preprint arXiv:2506.20790*, 2025.

Brianna Chrisman, Lucius Bushnaq, and Lee Sharkey. Identifying sparsely active circuits through local loss landscape decomposition. *arXiv preprint arXiv:2504.00194*, 2025.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2304.14997.

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. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy$_m$odel/$index.html$.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2024.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

Lennart Finke, Chandan Sreedhara, Thomas Dooms, Mat Allen, Emerald Zhang, Juan Diego Rodriguez, Noa Nabeshima, Thomas Marshall, and Dan Braun. Parameterized synthetic text generation with SimpleStories. *arXiv preprint arXiv:2504.09184*, 2025.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelsey Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. The quest for the right mediator: Surveying mechanistic interpretability for NLP through the lens of causal mediation analysis. *Computational Linguistics*, 2025. doi: 10.1162/COLI.a.572. URL https://direct.mit.edu/coli/article/doi/10.1162/COLI.a.572.

James Oldfield, Shawn Im, Yixuan Li, Mihalis A. Nicolaou, Ioannis Patras, and Grigorios G. Chrysos. Towards interpretability without sacrifice: Faithful dense layer decomposition with mixture of decoders. In *Advances in Neural Information Processing Systems*, 2025.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *Proceedings of the International Conference on Learning Representations*, 2023. arXiv:2211.00593.

A CLUSTERING DETAILS

Algorithm 1 Cross-Layer Mechanism Discovery

Require: SPD decomposition $\{(U_c^l, V_c^l, g_c^l)\}$, dataset $X = \{x_m\}_{m=1}^M$, k neighbors
Ensure: Mechanism clusters $\{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$
Step 1: Compute activation profiles and correlations
$A_i \leftarrow [g_i(x_1), \ldots, g_i(x_M)]$ for all subcomponents i
$\rho_{ij} \leftarrow \text{corr}(A_i, A_j)$ for all pairs (i, j)
Step 2: Build sparse similarity graph
$s_{ij} \leftarrow \max(0, \rho_{ij})$ ▷ clamp negative correlations to zero
for each subcomponent i **do**
    Keep only k largest s_{ij} values
end for
Symmetrize via mutual k-NN (keep edge (i, j) only if both nodes include each other)
Step 3: Spectral clustering
Compute degree matrix D with $D_{ii} = \sum_j s_{ij}$
Solve generalized eigenproblem $Lv = \lambda Dv$ where $L = D - S$
$K \leftarrow \arg\max_k(\lambda_{k+1} - \lambda_k)$ ▷ eigengap heuristic
Form embedding matrix $V \in \mathbb{R}^{N \times K}$ from first K eigenvectors
$\{\mathcal{M}_1, \ldots, \mathcal{M}_K\} \leftarrow k$-means(rows of V)
return Clusters $\{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$

Complexity The overall complexity of our method is dominated by three operations. First, computing the full $N \times N$ correlation matrix from activation profiles requires $O(MN^2)$ time, where M is the number of samples and N is the total number of subcomponents. Second, constructing the k-NN graph requires $O(N^2 \log k)$ time if we sort neighbors for each node. Third, spectral decomposition of the sparse graph Laplacian to extract K eigenvectors takes $O(kNK \cdot T_{\text{iter}})$ time. The final k-means clustering on the $N \times K$ embedding matrix requires $O(NKT_{km})$ time, where T_{km} is the number of k-means iterations (typically small and independent of N). In practice, for $N \in [50, 500]$ (typical for SPD decompositions of small language models).We assume this complexity would be beneficial when applying the same setting to larger language models.

Configuration For all experiments, we use the following clustering hyperparameters. For the activation profile dataset, we use $M = 1000$ samples from the validation set for toy models and $M = 5000$ sequences (10–100 tokens each) from held-out data for language models. For the k-NN graph, we set $k = 10$ nearest neighbors, as we found $k \in [10, 15]$ to be robust across settings. For the cluster count, the eigengap heuristic provides an initial K, which we then manually inspect for cluster semantics and merge or split as needed for interpretability.

B SPD DECOMPOSITION PROTOCOL

For all models, we follow the SPD training protocol of Bushnaq et al. (2025). Each targeted weight matrix is decomposed into C rank-one subcomponents, where we vary $C \in \{50, 100, 200, \ldots\}$ depending on model size. Gate networks are 2-layer MLPs with hidden dimension 64 and ReLU activations. Training uses the Adam optimizer with learning rate 10^{-3} to 5×10^{-4} and batch size 32–128. The loss combines reconstruction fidelity (MSE or cross-entropy), KL divergence between original and reconstructed outputs, and importance-minimality penalty weighted by λ_{sparse}. Training runs for 10,000–50,000 steps depending on model size and convergence.

We tune λ_{sparse} to explore the fidelity–sparsity trade-off: higher values produce sparser decompositions with cleaner importance profiles but higher reconstruction error, while lower values preserve fidelity at the cost of dense, noisier decompositions.

After training, we identify *active* components as those with maximum importance > 0.05 across the evaluation dataset, discarding components that never activate. This filtering is essential for language models where many initialized components remain dead throughout training.

C Implementation Details

Experiments are implemented in PyTorch 2.0 and run on NVIDIA A100 GPUs. SPD training takes 1–4 hours depending on model size; clustering on pre-trained decompositions takes < 1 minute for $N \leq 100$ components and $\sim$5 minutes for $N \approx 2000$. For toy models we report mean $\pm$ std over 5 random seeds; for language models we report single representative runs as mechanism discovery does not require averaging.

D Threshold Selection for Active Components

SPD produces importance values $g_c^l(x) \in [0, 1]$ for each subcomponent on each input. We classify a subcomponent as *alive* if $\max_x g_c^l(x) > \tau$ for a threshold τ. The threshold controls the trade-off between retaining potentially relevant components and filtering noise from near-zero importance values. We select τ based on the importance distribution of each model.

$\text{TMS}_{\text{5-2+ID}}$ ($\tau = 0.1$): This smaller model with an inserted identity matrix exhibits similar bimodal behavior. The threshold yields 36 active components, consistent with the expected $m_2 + m_1 = 5 + 2 = 7$ ground-truth mechanisms distributed across both layers (with multiple subcomponents per mechanism due to the identity pathway).

$\text{TMS}_{\text{40-10}}$ ($\tau = 0.1$): The strong sparsity regularization during SPD training produces a bimodal importance distribution—components are either nearly always inactive ($\max g < 0.05$) or clearly active ($\max g > 0.5$). A threshold of 0.1 cleanly separates these populations, yielding exactly 80 alive components.

Two-Layer Residual MLP ($\tau = 0.2$): This model exhibits a more distributed importance profile due to computation in superposition—neurons are used polysemantically across multiple input features, leading to moderate rather than binary importance values. A higher threshold of 0.2 filters components that activate only weakly, yielding 251 alive components (84.3% dead rate) which aligns with the expected 100 feature-computing mechanisms plus 50 identity projection subcomponents.

Three-Layer Residual MLP ($\tau = 0.1$): The deeper architecture spreads computation across more layers, resulting in lower per-component importance values on average. A threshold of 0.1 yields 379 alive components, close to the theoretical prediction of $102 \times 3 + 17 \times 3 = 357$ (102 features across 3 mlp_in layers plus 17 neurons per mlp_out layer).

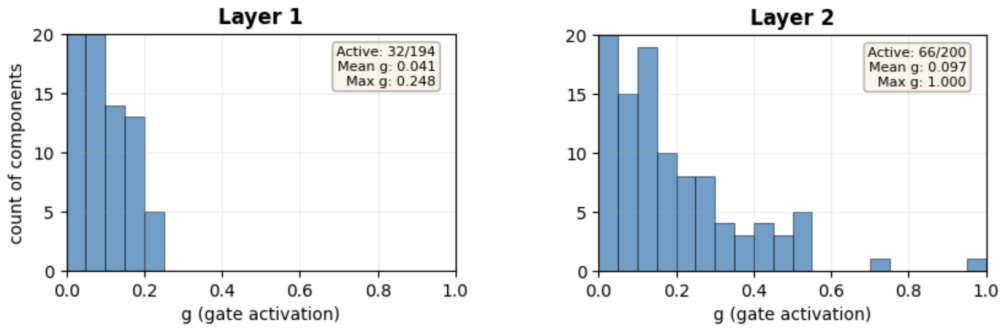
SimpleStories ($\tau = 0.1$): The larger transformer model with 56,000 total subcomponents shows a heavy-tailed distribution where 85% of components never exceed 0.1 on any input. We retain 8,425 active components (15.0% survival rate). Unlike toy models, no ground-truth mechanism count is available; we select $\tau = 0.1$ for consistency with toy model experiments and because it produces a tractable number of components for clustering while filtering clearly inactive subcomponents.

E $\text{TMS}_{\text{5-2+ID}}$: Detailed Results

Cluster	Size	Layer 1	Layer 2
0	11	5 (9,11,13,15,17)	6 (5,6,7,10,11,13)
1	7	2 (1,18)	5 (0,4,8,12,14)
2	5	3 (2,10,12)	2 (1,18)
3	7	4 (3,6,7,16)	3 (2,3,17)
4	6	3 (5,8,19)	3 (9,15,16)

Table 2: Spectral clustering results for $\text{TMS}_{\text{5-2+ID}}$ (36 active components). Each row shows cluster ID, total size, and count with component indices from each layer. All clusters span both layers.

F $TMS_{40\text{-}10}$: Detailed Results



(a) Layer 1 (b) Layer 2

Figure 5: Layer-wise activation distributions

High Correlation Examples. Representative encoder-decoder pairs with their correlation coefficients:

$$\rho = 0.997: \quad \text{linear1:42} \leftrightarrow \text{linear2:118} \quad \text{(feature 0)}$$
$$\rho = 0.995: \quad \text{linear1:87} \leftrightarrow \text{linear2:203} \quad \text{(feature 1)}$$
$$\rho = 0.998: \quad \text{linear1:156} \leftrightarrow \text{linear2:71} \quad \text{(feature 2)}$$

Feature Alignment Validation. Each cluster's encoder and decoder subcomponents maximize on the same feature:

Cluster	Layer 1	Layer 2	Maximized Feature
0	42	118	0
1	87	203	1
7	31	165	7
23	198	54	23
39	12	89	39

Table 3: Cross-layer feature alignment in $TMS_{40\text{-}10}$: subcomponents from encoder (Layer 1) and decoder (Layer 2) within each cluster maximize the same feature.

G Two-Layer Residual MLP: Detailed Results

Cross-Layer Pair Clusters. Spectral clustering identifies 100 pair clusters, each containing one mlp_in subcomponent from layer 0 and one from layer 1.

Cluster	Layer 0	Layer 1	Correlation (ρ)
12	47	83	1.000
37	112	156	1.000
64	203	89	1.000

Table 4: Representative cross-layer pair clusters in $TMS_{40\text{-}10}$, showing mlp_in subcomponents from consecutive layers with perfect correlation.

H THREE-LAYER RESIDUAL MLP: DETAILED RESULTS.

The three-layer variant uses 51 neurons across 3 layers (17 each) to compute 102 input-output functions.

Module	Alive	Dead	Expected
layers.0.mlp_in	102	398	102
layers.1.mlp_in	102	398	102
layers.2.mlp_in	124	376	102
layers.0.mlp_out	17	483	17
layers.1.mlp_out	17	483	17
layers.2.mlp_out	17	483	17

Table 5: Alive component counts per module in the three-layer residual MLP. The mlp_in layers show $\sim$102 alive components matching the 102 input features.

Representative Triplet Clusters. Each triplet activates maximally on inputs corresponding to the same feature index, confirming cross-layer mechanism discovery.

Cluster	Layer 0	Layer 1	Layer 2
27	99	125	131
40	105	127	147
22	169	66	229
30	63	96	139
53	144	91	0
85	331	435	390

Table 6: Representative triplet clusters showing mlp_in subcomponents from three consecutive layers that activate maximally on the same feature.

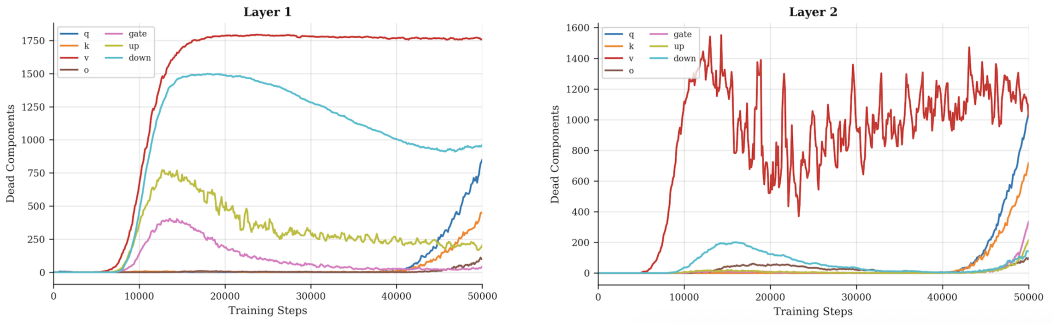
I SIMPLESTORIES: DETAILED RESULTS

Component Statistics by Module Type and Layer Distribution. Table 7 shows the average number of active components per layer for each module type and their survival rates.

Module Type	Active (avg/layer)	Survival Rate
mlp.down_proj	412	20.6%
mlp.gate_proj	398	19.9%
mlp.up_proj	385	19.3%
self_attn.q_proj	156	7.8%
self_attn.k_proj	148	7.4%
self_attn.v_proj	289	14.5%
self_attn.o_proj	223	11.2%

Table 7: Average active component counts per layer by module type in SimpleStories.

Dead Component Analysis. Figure 6 shows the distribution of dead components across different module types in layers 1 and 2 of SimpleStories.



(a) Layer 1 modules

(b) Layer 2 modules

Figure 6: Dead component distribution across layer modules in SimpleStories