Disentangling MLP Neuron Weights in Vocabulary Space

Asaf Avrahamy **Yoav Gur-Arieh** **Mor Geva**
Blavatnik School of Computer Science and AI, Tel Aviv University
{asafavrahamy@mail, yoavgurarieh@mail, morgeva@tauex}.tau.ac.il

Abstract

Interpreting the information encoded in language model weights remains a fundamental challenge in mechanistic interpretability. In this work, we introduce ROTATE (Rotation-Optimized Token Alignment in weighT spacE), a data-free method requiring no forward passes that disentangles MLP neurons directly in weight space. Our approach relies on a key statistical observation: neurons that encode coherent, monosemantic concepts exhibit high kurtosis when projected onto the model's vocabulary. By optimizing rotations of neuron weights to maximize their vocabulary-space kurtosis, our method recovers sparse, interpretable directions which we name *vocabulary channels*. Experiments on Llama-3.1-8B-Instruct and Gemma-2-2B-it demonstrate that ROTATE consistently recovers vocabulary channels that are faithful to the neuron's behavior; ablating individual channels selectively disables corresponding input activations or the promotion of specific concepts. Moreover, aggregating channel-level descriptions yields comprehensive neuron descriptions that outperform optimized activation-based baselines by 2–3× in head-to-head comparisons. By providing a data-free decomposition of neuron weights, ROTATE offers a scalable, fine-grained building block for interpreting language models.

1 Introduction

One of the underexplored goals of mechanistic interpretability is inspecting the information encoded in language model (LM) weights. Targeting weights is particularly appealing as it allows examining the model independently of specific inputs or data distributions, which can introduce biases (Bolukbasi et al., 2021; Gao et al., 2025) or incur high computational costs. A key challenge in interpreting LM weights is finding the "right unit of analysis" (Mueller et al., 2025; Sharkey et al., 2025; Geiger et al., 2025). While prior work has made progress in identifying neurons that capture individual, coherent concepts (Geva et al., 2021; 2022; Dai et al., 2022) and attention heads that implement specific functions (Zheng et al., 2025; Elhelo & Geva, 2025), in most cases these components are polysemantic and encode multiple entangled concepts (Bolukbasi et al., 2021; Gurnee et al., 2023).

In this work, we tackle the challenge of polysemanticity by disentangling model weights, focusing on MLP neurons in LMs. First, we make a key observation: MLP neurons that strongly promote single, coherent concepts exhibit high kurtosis when their weights are projected into the model's vocabulary space. This suggests that kurtosis in vocabulary space—a measure of how heavy-tailed the distribution over vocabulary tokens is—can serve as a proxy for directions with monosemantic attributes. Based on this observation, we introduce ROTATE (Rotation-Optimized Token Alignment in weighT spacE), a data-free method requiring no forward passes through the model that disentangles MLP neuron weights into their constituent, human-interpretable components. Given a neuron weight vector $\mathbf{w} \in \mathbb{R}^d$, ROTATE learns rotation matrices $\{\mathbf{R_i}\}$, each rotating $\mathbf{w}$ to reveal a semantically privileged basis in weight space $\mathbf{v}_i := \mathbf{R}_i\mathbf{w}$ (see Figure 1). Rotations are learned by optimizing towards increased vocabulary space kurtosis, while penalizing deviations from $\mathbf{w}$. We call these discovered vectors $\{\mathbf{v}_i\}$ *vocabulary channels*, as they are projections of the original neuron that are aligned with the vocabulary basis of the model.

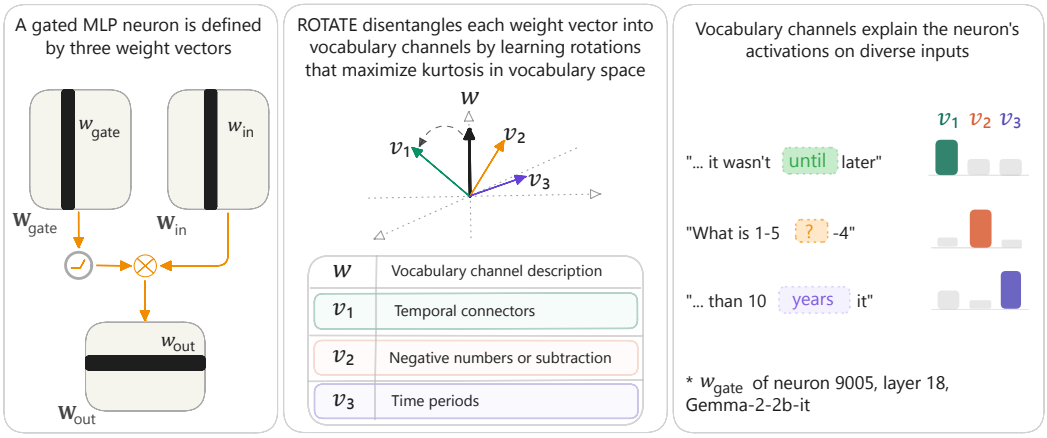

Figure 1: We propose to disentangle MLP neuron weights (**Left**) using ROTATE, a data-free method that learns rotations of a neuron's weight vector $\mathbf{w}$ to maximize kurtosis in the model's vocabulary space, recovering sparse, interpretable directions we call *vocabulary channels* (**Middle**). Each channel isolates a distinct concept encoded in $\mathbf{w}$, allowing a fine-grained understanding of the neuron's mechanism across diverse inputs (**Right**).

Through a series of experiments on Gemma-2-2B-it (Gemma Team et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), we show that vocabulary channels capture fine-grained functions that are faithful to the neuron's behaviors. Ablating individual channels selectively suppresses specific neuron functionalities without affecting others. Moreover, vocabulary channels provide more complete neuron explanations, covering a wider range of the neuron's activation space. Across both these evaluations, ROTATE outperforms decompositions by state-of-the-art sparse autoencoders (SAEs), Gemma Scope (Lieberum et al., 2024) and Llama Scope (He et al., 2024), applied to neuron weights. Next, we demonstrate the utility of ROTATE in generating natural-language neuron descriptions. By aggregating the descriptions of a neuron's channels, we produce descriptions that consistently outperform optimized descriptions over top-activating inputs (Choi et al., 2024) and a strong baseline that combines activating inputs with vocabulary projection (Gur-Arieh et al., 2025a), achieving 2–3× higher win rates in head-to-head comparisons across layers and evaluation sets.

In summary, our work makes the following contributions: (a) we observe that high-kurtosis vocabulary distributions correlate with monosemantic directions in LM weight space, (b) we introduce ROTATE, a data-free method that uses this signal for disentangling MLP weights into interpretable directions, (c) experiments on widely-used LMs show that ROTATE recovers faithful vocabulary channels that outperform SAE-based baselines on both faithfulness to neuron behavior and coverage of its activation spectrum, and (d) we show that aggregating vocabulary channels can produce better neuron descriptions than common automated interpretability approaches. We release our code at https://github.com/AsafAvr/rotating-neurons.

2 Preliminaries and notation

Neurons in LMs with gated MLP layers We focus on autoregressive transformer-based (Vaswani et al., 2017) LMs with a hidden dimension d and an inner MLP dimension d_a. Let $\mathbf{E} \in \mathbb{R}^{V \times d}$ and $\mathbf{U} \in \mathbb{R}^{d \times V}$ denote the embedding and unembedding matrices, where V is the vocabulary size. A gated MLP layer (Shazeer, 2020) is defined by three parameter matrices $\mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}}^T \in \mathbb{R}^{d_a \times d}$ and a nonlinear activation function σ:[1]

$$\text{MLP}(\mathbf{x}) = \mathbf{W}_{\text{out}} \left(\sigma(\mathbf{W}_{\text{gate}}\mathbf{x}) \odot (\mathbf{W}_{\text{in}}\mathbf{x}) \right) \tag{1}$$

[1]Our approach also can be applied to vanilla MLPs with only $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$.

where $\mathbf{x} \in \mathbb{R}^d$ is an input hidden state and $\odot$ denotes element-wise multiplication. A neuron is defined by an index $i \in [d_a]$ and acts as a computational unit with three weight vectors: **Input vectors** $\mathbf{w}^{(i)}_{\text{gate}}, \mathbf{w}^{(i)}_{\text{in}} \in \mathbb{R}^d$, which correspond to the i-th rows of $\mathbf{W}_{\text{gate}}$ and $\mathbf{W}_{\text{in}}$, respectively, and an **output vector** $\mathbf{w}^{(i)}_{\text{out}} \in \mathbb{R}^d$, corresponding to the i-th column of $\mathbf{W}_{\text{out}}$. The input vectors determine the neuron's activation pattern for a given input $\mathbf{x}$, while the output vector is written to the residual stream, weighted by the input's activation strength.

Vocabulary projection Projection to vocabulary space has been a common approach for analyzing model representations and weights (nostalgebraist, 2020; Geva et al., 2022; Dar et al., 2023). The projection $\mathbf{z} = \mathbf{wU}$ of a neuron's weight vector $\mathbf{w}$ yields a vector of logits $\mathbf{z} \in \mathbb{R}^V$, where the indices of the highest and lowest values in $\mathbf{z}$ correspond to the tokens that the neuron most strongly promotes or suppresses, respectively.

Kurtosis Kurtosis is the fourth standardized moment, which provides a statistical measure of the "tailedness" of a probability distribution. Here, we treat the logits $\mathbf{z} \in \mathbb{R}^V$ as a distribution over the vocabulary. A high kurtosis value indicates that the distribution is sharply peaked with heavy tails, meaning the neuron acts strongly on a sparse set of tokens while having little effect on most others. Thus, Gaussianity represents the "least interesting" distribution, and we maximize kurtosis to identify directions that are non-Gaussian, separating mixed signals into independent, sparse components. For the definition of kurtosis and an illustration, see §A.

3 High vocabulary kurtosis as a signal of monosemantic directions

To disentangle polysemantic neurons in weight space without ground-truth labels, we require an unsupervised measure that distinguishes interpretable, concept-centric directions from entangled or random ones. In this section, we identify vocabulary-projection kurtosis (vocabulary kurtosis in short), as such a signal. We ground this hypothesis with observations from prior work and validate it through empirical analysis.

Monosemantic neurons in LMs Prior work has identified neurons in LMs that strongly encode single, coherent concepts. Geva et al. (2022) showed that neuron weight vectors in $\mathbf{W}_{\text{out}}$ can be viewed as additive updates that promote the probability of a sparse set of semantically related tokens. More recently, Gurnee et al. (2024); Lad et al. (2024) identified a small set of "universal" neurons, characterized by high kurtosis in the vocabulary basis, that cluster densely in the middle-to-late layers during the "prediction ensembling" stage, suggesting that sparse, heavy-tailed distributions are a signature of output-facing computations. Last, Hong et al. (2025) found a set of MLP neurons called *concept vectors* in Llama-2-7B (Touvron et al., 2023) and OLMo-7B (Groeneveld et al., 2024), that exhibit monosemantic patterns in their vocabulary projections. These neurons strongly promote specific concepts, and ablating them degrades the model's ability to generate knowledge about the concepts they encode.

High kurtosis as a monosemanticity signal Given the above observations, we hypothesize that the distribution over the vocabulary induced by a weight vector could indicate how monosemantic it is. Specifically, we expect that monosemantic neurons will be correlated with higher kurtosis values of their vocabulary projections. To test this, we compare the vocabulary kurtosis values of the concept vectors found by Hong et al. (2025) with those of randomly sampled neurons of the same layers. Figure 2 shows that, for both Llama-2-7B and OLMo-7B, vocabulary kurtosis creates a clear separation between

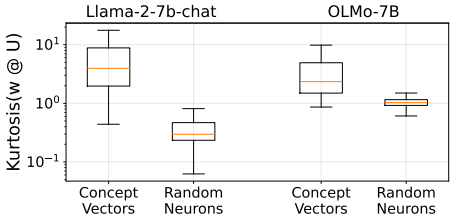

Figure 2: Vocabulary kurtosis of concept vectors in $\mathbf{W}_{\text{out}}$ (Hong et al., 2025) vs. random neurons from the same layers.

these groups of neurons. The median concept vector lies at the 90th percentile for Llama-2-7B and the 95th percentile for OLMo-7B relative to the randomly sampled neurons. As further validation of vocabulary kurtosis being a meaningful signal, we tracked its values during pre-training in OLMo-2-1124-7B (Walsh et al., 2025). Our analysis shows that vocabulary kurtosis rises sharply in early training and concentrates in middle and final layers — confirming it is a learned property rather than an artifact (see §B for details). Together, these observations motivate our approach: **low-kurtosis (polysemantic) neurons may be composed of multiple high-kurtosis (monosemantic) directions, which could be disentangled by maximizing non-Gaussianity**.

4 ROTATE

We now introduce ROTATE, a data-free method that, given a neuron weight vector $\mathbf{w}$, learns a set of rotation matrices $\{\mathbf{R_i}\}$, each yielding a *vocabulary channel* $\mathbf{v}_i := \mathbf{R_i}\mathbf{w}$ that describes a monosemantic direction of $\mathbf{w}$. An algorithm describing the method is provided in §C.

Optimization objective The core of our approach is in finding a rotation matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ such that the rotated vector $\mathbf{v} = \mathbf{w}\mathbf{R}$ will exhibit a high-kurtosis logit distribution $\mathbf{z} = \mathbf{v}\mathbf{U}$. To steer the optimization towards interpretable features while maintaining fidelity to the neuron, we minimize a loss function $\mathcal{L}$ composed of two competing terms: (a) **kurtosis loss** ($\mathcal{L}_{\text{kurt}}$), maximizing the kurtosis of $\mathbf{z}$ to push $\mathbf{w}$ towards monosemantic directions, and (b) **regularization loss** ($\mathcal{L}_{\text{reg}}$), penalizing the cosine distance between $\mathbf{v}$ and $\mathbf{w}$. This regularization anchors the discovered channels in $\mathbf{w}$, preventing convergence to arbitrary high-kurtosis directions.

$$\mathcal{L} = -\lambda \cdot \mathcal{L}_{\text{kurt}} + \mathcal{L}_{\text{reg}} = -\lambda \cdot \log(1 + \text{Kurt}(\mathbf{z})) + 1 - \frac{\mathbf{w}\cdot\mathbf{v}}{\|\mathbf{w}\|\|\mathbf{v}\|} \tag{2}$$

We minimize $\mathcal{L}$ via gradient descent over a Householder parameterization of $\mathbf{R}$ (Householder, 1958), which enforces orthogonality by construction. Let $\mathbf{h} \in \mathbb{R}^d$ be a learned vector, initialized as $\mathbf{h} \sim \mathcal{N}(0, I)$, we define $\mathbf{R}$ as:

$$\mathbf{R} = \mathbf{I} - 2\frac{\mathbf{h}\mathbf{h}^T}{\|\mathbf{h}\|^2} \tag{3}$$

This parameterization allows us to optimize a d-dimensional vector that creates a full rank reflection matrix. Notably, a single Householder matrix is technically a reflection, yet we find it sufficient (see details in §C.6 and §C.8 for method efficiency).

Iterative algorithm Optimizing Eq. 2 yields a single vocabulary channel. Since neurons often capture multiple concepts (Bricken et al., 2023; Scherlis et al., 2025; Gurnee et al., 2023), we apply the optimization iteratively. However, naively repeating independent runs converges to the same local optimum (§C.6), so we employ an iterative masking procedure.[2] After each iteration, we identify the tokens contributing most significantly to the channel's kurtosis and mask them to prevent re-discovery. Let $\mathbf{z} = \mathbf{v}\mathbf{U}$ be the logit vector of the discovered channel with mean $\mu_\mathbf{z}$ and standard deviation $\sigma_\mathbf{z}$. We mask high-contributing tokens with logit magnitudes exceeding k standard deviations:

$$\mathcal{T} = \{i : |z_i - \mu_\mathbf{z}| > k \cdot \sigma_\mathbf{z}\}, \tag{4}$$

This forces subsequent iterations to discover new high-kurtosis directions. We also mask known "glitch tokens" (Li et al., 2024; Land & Bartolo, 2024), which are under-trained embeddings whose extreme norms act as degenerate attractors (see §C.5). Each rotation $\mathbf{R}_i$ is optimized until loss convergence or a maximum step count.

[2] We also investigated other strategies but found token masking to be most consistent (see §C.6).

5 Experiments

A natural question that arises is whether the weight-derived directions by ROTATE capture the neuron's behavior during inference. To tackle this, we conduct evaluations along two axes: *faithfulness*, i.e., how accurately the discovered channels predict the neuron's activation patterns (input-side) and concept promotion (output-side), and *completeness*, i.e., how well the discovered channels explain the neuron's activation spectrum. We find that ROTATE's data-free channels obtain consistently higher faithfulness and completeness scores than data-driven SAE baselines, explaining a larger fraction of the neuron's behavior. Moreover, channel ablations causally affect the neuron's activations on specific examples, while preserving its activations on other examples. Additional evaluations of ROTATE show that it finds the same vocabulary channels across different initializations (see §C.4).

5.1 Experimental setup

The weight vectors $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ of a neuron can be viewed as "readers" from the residual stream and $\mathbf{w}_{\text{out}}$ as the "writer" (Geva et al., 2021). In our experiments, we apply ROTATE to $\mathbf{w}_{\text{gate}}$ for the input side and $\mathbf{w}_{\text{out}}$ for the output side, running $n_{\text{iter}} = 50$ iterations per weight vector which achieves high reconstruction (cosine similarity > 0.95, relative norm > 0.7), see §C.2 for analysis). We focus on $\mathbf{w}_{\text{gate}}$ rather than $\mathbf{w}_{\text{in}}$ for the input side as the gating activation is mostly positive, which simplifies the analysis, but ROTATE is equally applicable to $\mathbf{w}_{\text{in}}$. Hyperparameters are selected via grid search on a disjoint set of neurons (see §C.7 for details). Using this configuration, we apply ROTATE to Gemma-2-2B-it (Gemma Team et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We focus on two layers per model, selected for their alignment with the vocabulary basis:[3] in Gemma, which uses tied embeddings (i.e., $E = U^T$), we analyze an early and a middle layer (layers 4 and 18) where weight vocabulary projection is geometrically valid; in Llama, we evaluate middle-to-late layers (layers 18 and 22), where the residual stream is most aligned with the unembedding matrix (nostalgebraist, 2020; Geva et al., 2021; Lee et al., 2025). From each layer we sample 100 random neurons. Examples of obtained channels are provided in §D.

Let $\mathcal{C} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be the set of channels obtained for a neuron, given an input residual stream vector $\mathbf{x}$, we define the *top channel* as $\mathbf{v}^* := \arg\max_{\mathbf{v} \in \mathcal{C}} (\mathbf{x} \cdot \mathbf{v})$, i.e., the channel most aligned with $\mathbf{x}$.

Evaluation data To validate the behavior of the extracted channels during inference on inputs, we collect a dataset $\mathcal{D}$ of 2 million tokens from the Pile (Gao et al., 2020), recording each token's residual stream vector before the MLP layer and the corresponding neuron activations. This dataset is used in our experiments for retrieving top-activating examples and computing channel–example alignments.

Channel descriptions To evaluate channels, we first produce a natural-language description for each one. Following Gur-Arieh et al. (2025a), we prompt an LLM with two sources of evidence: the top-50 tokens in the channel's vocabulary projection and its top activating examples from $\mathcal{D}$ (see §G for the full prompt).

5.2 Input-side channel faithfulness

Following automated interpretability protocols (Bills et al., 2023; Choi et al., 2024; Paulo et al., 2025), we test whether the concept captured by a channel activates its corresponding neuron. Adopting the evaluation setup of Huang et al. (2023), given a channel description, we prompt an LLM to create two sets of examples: *activating* examples that match the description and *neutral* examples that do not. We then pass both sets through the model and record each neuron's maximum activation across token positions per example. This yields two sets of activation values per neuron $A_{\text{activating}}$ and A_{neutral}. A channel is considered

[3] Evaluations of ROTATE across early, middle, and late layers in both models show that it consistently scores above the random baseline (§C.3), with the strongest gains concentrated in middle-to-late layers, consistent with the training dynamics of vocabulary kurtosis (§B).

	Faithfulness				Completeness			
	Llama-3.1		Gemma-2		Llama-3.1		Gemma-2	
Method	$\ell = 18$	$\ell = 22$	$\ell = 4$	$\ell = 18$	$\ell = 18$	$\ell = 22$	$\ell = 4$	$\ell = 18$
ROTATE (Ours)	**0.71**	**0.58**	**0.46**	**0.47**	**0.55**	**0.49**	**0.55**	**0.60**
SAE-Neuronpedia	0.45	0.41	0.33	0.35	0.44	0.41	0.42	0.49
SAE-TopK	0.49	0.46	0.34	0.37	0.40	0.40	0.36	0.42
Random	0.25	0.20	0.17	0.24	0.20	0.20	0.20	0.20

Table 1: Average Faithfulness and Completeness scores. ROTATE consistently outperforms SAE-based baselines across models and layers. Random reflects chance-level performance.

faithful if $\mathbb{E}[a \in A_{\text{activating}}] > \mathbb{E}[a \in A_{\text{neutral}}]$, evaluated via a one-sided t-test ($p < 0.05$) with 40 samples in each set. Namely, the channel captures a concept that activates the neuron more strongly than other concepts.

As existing interpretability methods do not disentangle individual neuron weights into fine-grained components, we adapt Gemma Scope and Llama Scope SAEs (Lieberum et al., 2024; He et al., 2024) trained on residual stream activations. Concretely, we follow the weight-disentangling procedure of Gur-Arieh et al. (2025b), which decomposes neuron weights by selecting the top-k features of a trained residual-stream SAE most aligned with $\mathbf{w}$; this is done by computing the dot product between $\mathbf{w}$ and each feature vector in the SAE's encoder and selecting the top-k features with the highest alignment (see §E.1 for more details). These features serve as counterparts to ROTATE's vocabulary channels. We describe the selected features with two approaches, with their difference isolating the effect of the channel/feature discovery method from the description generation procedure:

- **SAE-Neuronpedia**: Descriptions from Neuronpedia (Lin & Bloom, 2023) produced by prompting GPT-4 (OpenAI et al., 2024) with each feature's top-activating examples.
- **SAE-TopK**: Descriptions generated using the same procedure applied to ROTATE channels (§5.1), collecting the top tokens from the feature's vocabulary projection and the top activating examples, then prompting an LLM to produce a description.

Table 1 presents the faithfulness scores, showing that ROTATE consistently outperforms the SAE baselines (0.46–0.71 vs. 0.33–0.49). The advantage is most pronounced in layer 18 of Llama-3.1 (0.71 vs. 0.49), likely because middle layers develop the strongest vocabulary-aligned structure (see analysis in §B), providing a richer signal for ROTATE's kurtosis-based optimization. In contrast, the gap narrows in layer 4 of Gemma-2 (0.46 vs. 0.34), where early-layer neurons may encode more distributed representations that are harder to disentangle. The gap between ROTATE and SAE-based methods suggests that weight-derived channels describe neuron activations more accurately than residual stream features extracted from SAEs. Notably, all methods substantially exceed the random baseline, confirming that both approaches capture meaningful structure, though ROTATE captures it more precisely.

In addition to these results, we also evaluated two weight-space baselines on Gemma-2: **TopK-Iter**, an iterative logit-lens decomposition that repeatedly takes and masks the top tokens of $\mathbf{wU}$, and **FastICA** (Hyvärinen & Oja, 2000) that shares ROTATE's kurtosis-deflation structure. Both baselines performed at or below chance; see §E.1 for details and results.

Causal validity via channel ablation To test whether channels are causally responsible for the neuron's activation, we ablate the channel $\mathbf{v}$ from the neuron's weight vector $\mathbf{w}$ by projecting out its contribution: $\mathbf{w}_{\text{ablated}} = \mathbf{w} - (\mathbf{w} \cdot \mathbf{v})\,\mathbf{v}$. Then, we compare the neuron activations before and after ablation. Intuitively, if the channel controls a specific part of the neuron's behavior, then removing it should suppress activations on inputs related to that channel, while leaving other activations intact.

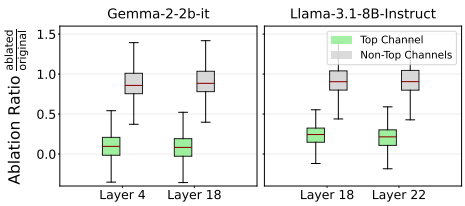

Figure 3: Input-side causal validity. Ablating the neuron's top channel drives its activation toward 0; ablating other channels leaves it near 1.

For each weight vector $\mathbf{w}$, we retrieve its top-1,000 activating examples from $\mathcal{D}$ and assign each example $\mathbf{x}$ to its top channel $\mathbf{v}^*$ (see §5.1). Then, we ablate $\mathbf{v}^*$ from $\mathbf{w}$ and compute the ablation ratio, defined as the ratio between the ablated neuron's activation and the original activation for $\mathbf{x}$. We measure this ratio on two sets of examples: those assigned to $\mathbf{v}^*$ and those assigned to other channels.

Figure 3 shows that ablating the activated channel drives the ratio toward 0 (**green**), confirming that the channel is responsible for the neuron's firing on those inputs. Ablating a non-activated channel leaves the ratio near 1 (**gray**), indicating that different channels do not interfere with one another. This shows that the discovered channels are both causally relevant and well-separated, with each governing a distinct subset of the neuron's behavior.

5.3 Output-side channel faithfulness

While input-side channels are selectively activated by different inputs, output-side channels all contribute simultaneously when the neuron fires. Thus, to evaluate faithfulness of output-side channels, we test what concepts the neuron promotes and whether ablating certain channels removes the expression of their concepts through the neuron.

We apply channel ablation as in §5.2, now targeting channels in $\mathbf{w}_{\text{out}}$. To assess the effect of ablating a channel $\mathbf{v}$, we leverage the Patchscopes framework (Ghandeharioun et al., 2024) to decode information from $\mathbf{w}_{\text{out}}$ and the ablated vector $\mathbf{w}_{\text{ablated}}$. Specifically,

we feed to the model: `"cat → cat; 135 → 135; hello → hello;"` followed by either $\mathbf{w}_{\text{out}}$ or $\mathbf{w}_{\text{ablated}}$. The few-shot format and conditioning the generation on the weight vector push the model to decode information from it. Now, let $T_{\mathbf{v}}$ denote the set of top-50 tokens in the vocabulary projection of the channel $\mathbf{v}$. We decode each of $\mathbf{w}_{\text{out}}$ and $\mathbf{w}_{\text{ablated}}$ multiple times, pooling all generated tokens per vector. Then, we compute the fraction of decoded tokens that belong to $T_{\mathbf{v}}$ in each pool, denoted f_{out} and f_{ablated}, respectively, and report the relative change $\Delta = (f_{\text{ablated}} - f_{\text{out}})/f_{\text{out}}$. For more details, see §E.4. We compare two ablations: *self-channel ablation*, where we ablate the channel whose token set $T_{\mathbf{v}}$ we monitor, and *cross-channel ablation*, where we ablate a different channel from the same neuron. If the channels are causally disentangled, self-channel ablation should suppress the channel's tokens while cross-channel ablation should leave them intact.

Table 2 presents the results. Self-channel ablation leads to near-complete suppression of the corresponding tokens (from -87% to -90%). In contrast, cross-channel ablation slightly increases the frequency (from $+14\%$ to $+24\%$), suggesting that a channel's tokens become more prominent when competing channels are removed. This confirms that the discovered output channels are causally separated; each independently controls its corresponding concept, and removing one does not collapse the neuron's other functions.

Model	Layer	Self (%)	Cross (%)
Gemma-2	4	-90 ± 34	$+24 \pm 55$
-2b-it	18	-87 ± 40	$+24 \pm 57$
Llama-3.1	18	-90 ± 35	$+15 \pm 60$
-8B	22	-88 ± 37	$+14 \pm 60$

Table 2: Output-side causal validity via channel ablation. Mean ($\pm$ std) % change in token frequency after self- or cross-channel ablation.

5.4 Decomposition completeness

The previous evaluations focused on whether a channel faithfully captures the behavior of its neuron. A question that remains is how many of the neuron behaviors do channels cover. We approach this by evaluating *completeness*, measuring how well the set of discovered channels collectively explains the neuron's activation landscape. Specifically, we focus on input-side channels in $\mathbf{W}_{\text{gate}}$ which admit a natural test: given diverse inputs that activate the neuron, can we match each to an appropriate channel?[4]

[4]Output-side channels lack this structure; when a neuron activates, it promotes all its output channels, making it unclear how to attribute individual activations to specific channels.

For every gate weight vector, we retrieve a sample of 100 out of its top-1000 activating input texts from $\mathcal{D}$ and, for each input t, identify its activated channel $\mathbf{v}^*$ (as defined in §5.1). We then assess whether the description of $\mathbf{v}^*$ explains the neuron activation on t, for every such input-channel pair. Using Gemini-3.1-Flash-Lite (Google, 2025) as an LLM judge (see validation in §E.5), we present the input text corresponding to $\mathbf{x}$ alongside five candidate channel descriptions: the description of $\mathbf{v}^*$ and four distracting descriptions sampled from channels of other neurons. The judge selects which description best explains why the neuron activated on this input. We report *matching accuracy*, defined as the fraction of examples where the judge selects the matched channel. The full judge prompt and an example query are provided in §E.3. We compare ROTATE channels against random channels of other neurons, establishing a random baseline of 20%, and the SAE-Neuropedia and SAE-TopK baselines from §5.2.

Table 1 presents the completeness scores. Across models and layers, ROTATE consistently outperforms the SAE baselines, achieving a matching accuracy of 49%–60% compared to 36%–49% for SAE features, both well above the 20% chance level. For more than half of the neuron's top activating inputs, an LLM judge can correctly identify corresponding ROTATE channel descriptions to the input, indicating that the discovered channels collectively cover the majority of the neuron's top activations.

6 Enhancing neuron descriptions

In this section, we show that vocabulary channels can be leveraged to produce more comprehensive textual descriptions of neuron activations compared to existing pipelines.

Description generation ROTATE produces dozens of channels per weight vector, raising the question of how to aggregate them into a single, coherent neuron description. Here, we experimented with four strategies, aggregating the descriptions of the first 25 channels from each of $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ (channel descriptions were obtained as in §5.2). From these strategies, we selected the following polarity-aware approach via a pairwise evaluation (see §F for details and results for all variants). This approach exploits the distinct roles of the two weight vectors in the gated MLP: $\mathbf{w}_{\text{gate}}$ controls whether the neuron fires and $\mathbf{w}_{\text{in}}$ determines the activation's sign. We split $\mathbf{w}_{\text{in}}$ channels by their vocabulary projection skewness polarity and pair each group with all gate channels, yielding two per-neuron descriptions: one for positive and one for negative activations, each synthesized by Gemini-2.0-Flash (see §F.3). Results below are from both polarities.

Baselines We compare ROTATE-based descriptions against prominent baselines:

- **MaxAct+VocabProj**: We collect the neuron's 20 top-activating inputs from the Pile (Gao et al., 2020) and concatenate them with the top-50 vocabulary tokens in the projections of $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$. Then, we prompt Gemini-2.0-Flash to generate a concise description (see §F for the full prompt). This approach has been shown to outperform descriptions based on each source alone (Gur-Arieh et al., 2025a).
- **MaxAct++**: As the strongest activation-based baseline, we use the descriptions by Choi et al. (2024) for neurons in Llama-3.1-8B-Instruct. These descriptions were generated via a multi-stage pipeline that involves the generation of candidate descriptions from top-activating inputs and scoring by a simulator that predicts per-token activations from a description. These automated descriptions have been shown to surpass human annotations on automated metrics.

Description evaluation We evaluate on 150 random neurons from Llama-3.1-8B-Instruct across 3 layers: 18 and 22 as in §5 and additionally layer 12 to test how the method performs in earlier layers. To evaluate their descriptions in head-to-head comparisons we use Gemini-3-Flash (Google, 2025) as a judge (see §E.5 for validation). Given an activating example and two candidate descriptions, the judge selects which description better explains the activation. To control for position bias, we run each comparison twice with swapped order. We declare a winner when both orderings agree and otherwise a tie. We evaluate descriptions

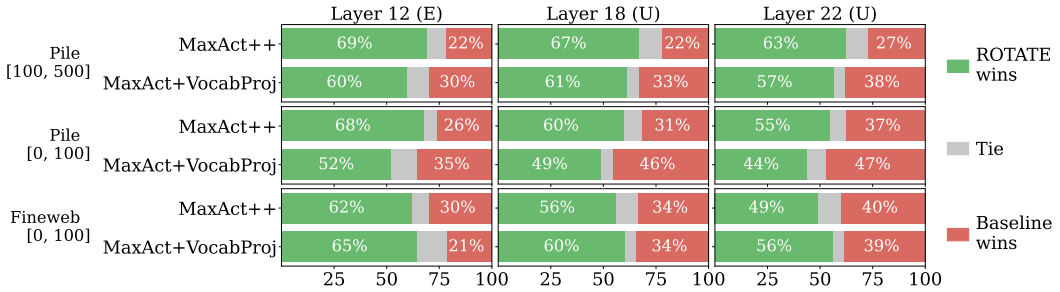

Figure 4: Head-to-head pairwise evaluation of ROTATE vocabulary channel descriptions against MaxAct+VocabProj and MaxAct++ baselines on Llama-3.1-8B-Instruct. Each bar shows the fraction of comparisons won by **ROTATE**, tied, or won by the **baseline**. Columns correspond to layers; rows to evaluation data sources and activation-rank ranges.

on three setups: (a) *top 100 Pile activating inputs*, testing if descriptions capture the neuron's most pronounced behavior; (b) *top 100-500 Pile activating inputs*, testing coverage beyond peak behavior; and (c) *top 100 FineWeb activating inputs*, drawn from the MaxAct++ held-out test set (Penedo et al., 2024), testing generalization to a different data distribution. Pile evaluation examples are drawn from a disjoint subset not used for description generation.

Results Figure 4 shows the results, and examples are given in §F.4. ROTATE wins against both baselines across nearly all setups. Against MaxAct++ the largest margins appear on moderate Pile activations (ranks 100–500), where ROTATE achieves 63%–69% win rates, where MaxAct++ is furthest from its top-activation training regime. Against MaxAct+VocabProj, wins are most pronounced on the same moderate (ranks 100–500) range and on FineWeb (a different data distribution) while on top Pile activations the two methods are nearly tied. This reflects a basic trade-off: activation-based methods condition on extreme responses, giving strong signal for peak behavior but limited coverage elsewhere, whereas ROTATE decomposes the weight vector independently of activation regime, naturally capturing concepts that surface at moderate levels. These results demonstrate the practical gains of weight-derived vocabulary channels for neuron-level interpretability.

7 Related work

Prior work has interpreted the weights of MLP layers (Geva et al., 2021; 2022) and attention heads (Elhage et al., 2021; Dar et al., 2023; Elhelo & Geva, 2025) in the vocabulary space. We build on this framework and learn rotations that disentangle neuron weights into monosemantic components. Other works have identified underlying structures in MLP weights; Adler et al. (2025) showed that MLPs in small networks can pack features via combinatorial "feature channel codes", Pearce et al. (2025) found that bilinear MLPs can admit eigen-decomposition of their weights into interpretable components, and Shafran et al. (2025) used MLP activations to discover neuron combinations that capture concepts and outperform SAEs on causal steering. Unlike these works, ROTATE achieves *data-free* decomposition of MLP layers in modern LMs.

Our study also relates to a large body of work on neurons in LMs (Sajjad et al., 2022), and contributes to tackling the challenge of polysemanticity (Elhage et al., 2022; Arora et al., 2018; Gurnee et al., 2023). While SAEs have been the dominant approach to recovering monosemantic units in LMs (Bricken et al., 2023; Huben et al., 2024; Gao et al., 2025), they require large-scale activation data. Recently, Gur-Arieh et al. (2025b) adapted residual-stream SAEs to decompose neuron weights. We compare against this approach and show that ROTATE consistently outperforms it in faithfulness and completeness with respect to the neuron's behavior. ROTATE also complements efforts to automatically describe neurons (Bills et al., 2023; Choi et al., 2024; Shaham et al., 2024; Gur-Arieh et al., 2025a) by leveraging their fine-grained decompositions into channels.

Several methods discover interpretable structure by optimizing orthogonal transformations. DAS (Geiger et al., 2024) learns such matrices via supervised gradient descent to isolate causal features in the residual stream, and Huang & Hahn (2026) learn them without supervision, partitioning activation space into interpretable subspaces by minimizing within-subspace nearest-neighbor distances over a large corpus of activations. ROTATE learns similar rotations, but without supervision or data, operating entirely in weight space.

Key components of ROTATE—kurtosis maximization, an orthogonality constraint, and iterative deflation—mirror the structure of classical Independent Component Analysis (ICA; Comon, 1994) and Projection Pursuit (Friedman & Tukey, 1974), in particular deflation-based algorithms such as FastICA (Hyvärinen & Oja, 2000) and RobustICA (Zarzoso & Comon, 2010) that extract non-Gaussian components one at a time. The key difference lies in what the statistic is computed over. Classical ICA estimates an unmixing matrix from a data matrix of many observations, measuring non-Gaussianity across samples, whereas ROTATE has no observations: it operates on a single static weight vector and measures kurtosis across the V vocabulary logits of its projection through the fixed unembedding $\mathbf{U}$, removing exactly the data dependence that both DAS and ICA require.

8 Conclusion and discussion

We introduce ROTATE, a data-free method that disentangles MLP neuron weights into interpretable vocabulary channels by maximizing kurtosis in the model's vocabulary space. The discovered channels provide faithful, causally meaningful descriptions of neuron behavior, outperforming SAE-based baselines in terms of faithfulness and completeness. Moreover, aggregating channel descriptions yields comprehensive neuron descriptions that achieve higher win rates over existing approaches. Taken together, vocabulary channels are positioned as a scalable, fine-grained unit of analysis for interpreting LMs. Future work could leverage ROTATE for more accurate, fine-grained circuit discovery and for studying interactions between network components. Further discussion on limitations is in §C.10.

Acknowledgments

We thank Ori Yoran and Guy Dar for valuable feedback, and Or Shafran, Clara Suslik, Daniela Gottesman, and Shir Rashkovits for their help with the evaluation of the LLM judge. This research was supported in part by the Academic Research Program at Google, Len Blavatnik and the Blavatnik Family foundation, the Alon Scholarship, a Coefficient Giving grant, and the Israel Science Foundation grant 1083/24.

References

Micah Adler, Dan Alistarh, and Nir Shavit. Towards combinatorial interpretability of neural computation. *arXiv [cs.LG]*, April 2025.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034. URL https://aclanthology.org/Q18-1034/.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, May 2023.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for BERT. *arXiv [cs.CL]*, April 2021.

Trenton 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. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Nitay Calderon, Roi Reichart, and Rotem Dror. The alternative annotator test for LLM-as-a-judge: How to statistically justify replacing human annotators with LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16051–16081, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.782. URL https://aclanthology.org/2025.acl-long.782/.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description. https://transluce.org/neuron-descriptions, October 2024.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3): 287–314, 1994. ISSN 0165-1684. doi: https://doi.org/10.1016/0165-1684(94)90029-9. URL https://www.sciencedirect.com/science/article/pii/0165168494900299. Higher Order Statistics.

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

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16124–16170, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.893. URL https://aclanthology.org/2023.acl-long.893/.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

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. *arXiv [cs.LG]*, September 2022.

Amit Elhelo and Mor Geva. Inferring functionality of attention heads from their parameters. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17701–17733, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.866. URL https://aclanthology.org/2025.acl-long.866/.

J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.*, 23(9):881–890, September 1974. ISSN 0018-9340. doi: 10.1109/T-C.1974.224051. URL https://doi.org/10.1109/T-C.1974.224051.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800GB dataset of diverse text for language modeling. *arXiv [cs.CL]*, December 2020.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

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

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-Yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M R Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size. *arXiv [cs.CL]*, July 2024.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3/.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: a unifying framework for inspecting hidden representations of language models.

In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Google. A new era of intelligence with Gemini 3, 2025. URL https://blog.google/products/gemini/gemini-3. Accessed: 2025-02-01.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer,

Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. *arXiv [cs.AI]*, July 2024.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.841.

URL https://aclanthology.org/2024.acl-long.841/.

Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. Enhancing automated interpretability with output-centric feature descriptions. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5757–5778, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.288. URL https://aclanthology.org/2025.acl-long.288/.

Yoav Gur-Arieh, Clara Haya Suslik, Yihuai Hong, Fazl Barez, and Mor Geva. Precise in-parameter concept erasure in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18986–19006, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.960. URL https://aclanthology.org/2025.emnlp-main.960/.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=JYs1R9IMJr.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ZeI104QZ8I.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, et al. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*, 2024.

Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. Intrinsic test of unlearning using parametric knowledge traces. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19524–19546, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.985. URL https://aclanthology.org/2025.emnlp-main.985/.

Alston S. Householder. Unitary triangularization of a nonsymmetric matrix. *J. ACM*, 5 (4):339–342, October 1958. ISSN 0004-5411. doi: 10.1145/320941.320947. URL https://doi.org/10.1145/320941.320947.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 317–331, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.24. URL https://aclanthology.org/2023.blackboxnlp-1.24/.

Xinting Huang and Michael Hahn. Decomposing representation space into interpretable subspaces with unsupervised learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SI2OZJuvPO.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00026-5.

Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of LLMs: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=R5unwb9KPc.

Sander Land and Max Bartolo. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11631–11646, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.649. URL https://aclanthology.org/2024.emnlp-main.649/.

Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=aJDykpJAYF.

Yuxi Li, Yi Liu, Gelei Deng, Ying Zhang, Wenjia Song, Ling Shi, Kailong Wang, Yuekang Li, Yang Liu, and Haoyu Wang. Glitch tokens in large language models: Categorization taxonomy and effective detection. *Proc. ACM Softw. Eng.*, 1(FSE), July 2024. doi: 10.1145/3660799. URL https://doi.org/10.1145/3660799.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://aclanthology.org/2024.blackboxnlp-1.19/.

Johnny Lin and Joseph Bloom. Neuronpedia: Interactive reference and tooling for analyzing neural networks with sparse autoencoders, 2023. URL https://neuronpedia.org. Software available from neuronpedia.org.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, et al. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, pp. 1–48, 2025.

nostalgebraist. interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, August 2020. Accessed: 2025-7-1.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade

Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EemtbhJOXc.

Michael Pearce, Thomas Dooms, Alice Rigg, Jose Oramas, and Lee Sharkey. Bilinear mlps enable weight-based mechanistic interpretability. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 47283–47310, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7504142a20a3e1fe9dd7de42f475828c-Paper-Conference.pdf.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv [cs.CL]*, June 2024.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep NLP models: A survey. *Trans. Assoc. Comput. Linguist.*, 10:1285–1303, November 2022.

Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks, 2025. URL https://arxiv.org/abs/2210.01892.

Or Shafran, Atticus Geiger, and Mor Geva. Decomposing mlp activations into interpretable features via semi-nonnegative matrix factorization, 2025. URL https://arxiv.org/abs/2506.10920.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=91H76m9Z94. Survey Certification.

Noam Shazeer. GLU variants improve transformer. *arXiv [cs.LG]*, February 2020.

Alessandro Stolfo, Ben Wu, Wes Gurnee, Yonatan Belinkov, Xingyi Song, Mrinmaya Sachan, and Neel Nanda. Confidence regulation neurons in language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 125019–125049. Curran Associates, Inc., 2024. doi: 10.52202/079017-3970. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e21955c93dede886af1d0d362c756757-Paper-Conference.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1288–1301, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.75. URL https://aclanthology.org/2024.findings-acl.75/.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU.

Vicente Zarzoso and Pierre Comon. Robust independent component analysis by iterative maximization of the kurtosis contrast with algebraic optimal step size. *IEEE Transactions on Neural Networks*, 21(2):248–261, 2010. doi: 10.1109/TNN.2009.2035920.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. Attention heads of large language models. *Patterns*, 6(2): 101176, 2025. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2025.101176. URL https://www.sciencedirect.com/science/article/pii/S2666389925000248.

A Additional preliminaries

A.1 Kurtosis and Skewness

Kurtosis is the fourth standardized moment of a distribution:

$$\mathrm{Kurt}(X) = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] - 3 \tag{5}$$

where μ and σ are the mean and standard deviation of X. We subtract 3 so that a Gaussian distribution has kurtosis zero (*excess kurtosis*). Positive values indicate heavier tails and a sharper peak than a Gaussian, meaning more of the variance is due to rare, extreme values.

Skewness is the third standardized moment, measuring the asymmetry of a distribution:

$$\mathrm{Skew}(X) = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{6}$$

Positive skewness indicates a heavier right tail (extreme positive logits dominate), while negative skewness indicates a heavier left tail (extreme negative logits dominate). In our setting, we use skewness polarity to distinguish channels that *promote* tokens (positive skewness) from those that *suppress* them (negative skewness).

In our setting, we treat the logit vector $\mathbf{z} = \mathbf{wU} \in \mathbb{R}^V$ as a distribution over the vocabulary: high kurtosis indicates that the neuron acts strongly on a sparse set of tokens while having negligible effect on the rest, and the skewness sign determines whether those tokens are promoted or suppressed. Figure 5 illustrates this contrast.

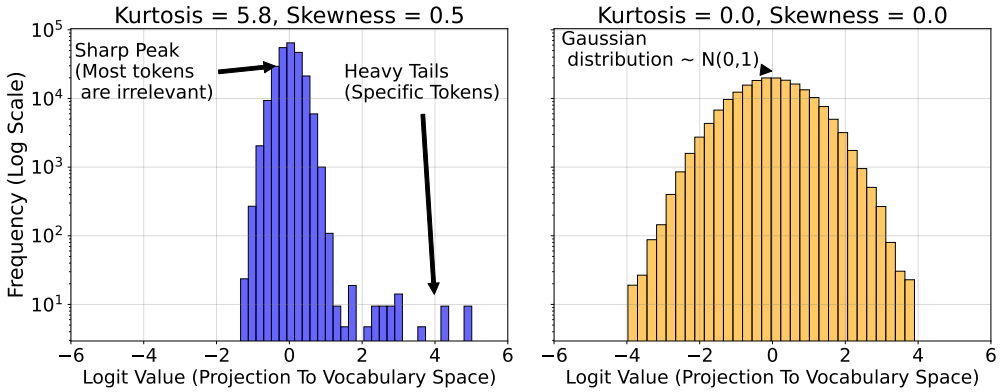

Figure 5: A distribution with high kurtosis and positive skewness, concentrated around zero with few extreme outliers (left), compared to a Gaussian (right).

B Vocabulary kurtosis across training and model families

Across training To verify that vocabulary kurtosis reflects genuinely learned structure rather than a static property of random initialization, we track its evolution during pre-training. Figure 6 shows the median vocabulary kurtosis of $\mathbf{W}_{\text{out}}$ neurons in OLMo-2-1124-7B (Walsh et al., 2025) across 4 trillion training tokens. At initialization, kurtosis

values are near zero (consistent with Gaussian-distributed weights). During early training, median kurtosis rises sharply before stabilizing, with the strongest concentration emerging in middle layers (around layers 15–20) and the final layers. This temporal and layer-wise pattern confirms that vocabulary-aligned monosemantic structure is actively shaped by training.

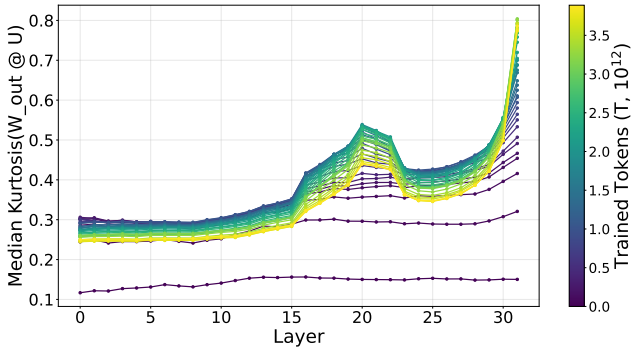

Figure 6: Median vocabulary kurtosis values of neuron weights in $\mathbf{W}_{\text{out}}$ across layers and checkpoints of OLMo-2-1124-7B (Walsh et al., 2025). We observe clear learning dynamics, rising sharply in early training and concentrating in middle and late layers. This temporal pattern confirms that vocabulary-aligned monosemantic structure is a learned property.

Across model families This layer-wise pattern, where middle-late and output-facing layers develop the strongest vocabulary-aligned structure, is consistent across multiple model families, as can be seen in Figure 7.

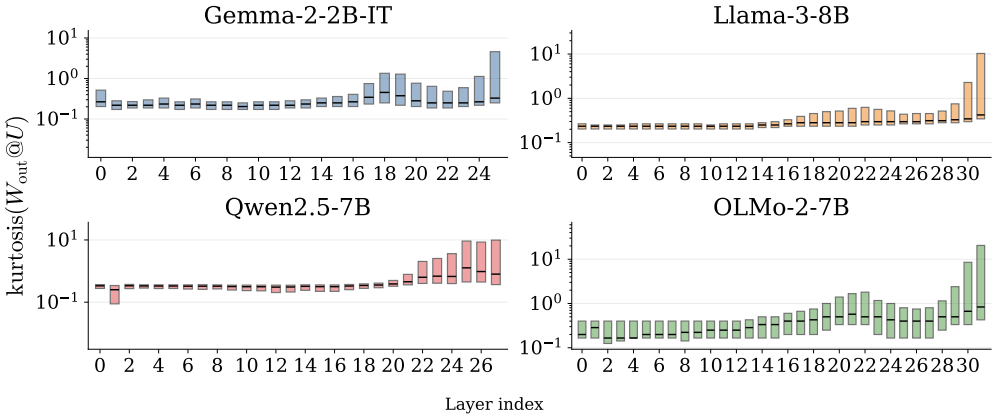

Figure 7: Per-layer vocabulary kurtosis distributions of $\mathbf{W}_{\text{out}}$ neurons for one representative model per family.

C ROTATE additional details and results

C.1 Algorithm

Algorithm 1 provides the full pseudo-code for ROTATE. Given a neuron weight vector $\mathbf{w}$ and the unembedding matrix $\mathbf{U}$, the method iteratively discovers vocabulary channels by optimizing Householder reflections to maximize vocabulary-space kurtosis. Each iteration yields a single channel; after discovery, the tokens driving its kurtosis are masked to force subsequent iterations toward new directions. The process terminates after n_{iter} iterations. Below we provide additional details on implementation choices and design decisions.

Algorithm 1 ROTATE

Input: MLP weight vector $\mathbf{w}$, unembedding matrix $\mathbf{U}$, kurtosis function $\gamma(x)$, kurtosis threshold τ, learning rate η, λ, standard deviation magnitude k, n_{iter}, n_{step}.
Output: Set of discovered rotation matrices $\mathcal{R}$.
$\mathbf{m} \leftarrow$ `init_mask(`$\mathbf{U}$`)`
$\mathcal{R} \leftarrow \{\}, \quad i \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  $\mathbf{h} \sim \mathcal{N}(0, I)$ ▷ Random initialization
  $\mathbf{R} \leftarrow I - 2\frac{\mathbf{h}\mathbf{h}^T}{\|\mathbf{h}\|^2}$ ▷ Householder reflection
  `optimizer` $\leftarrow$ `AdamW(`η`)`
  $s \leftarrow 0$
  while $s < n_{\text{step}}$ **do**
    $\mathbf{v} \leftarrow \mathbf{w}\mathbf{R}$ ▷ Rotate $\mathbf{v}$ with R
    $\mathbf{z} \leftarrow \mathbf{v}\mathrm{U}$ ▷ Obtain logits vector
    $\hat{\mathbf{z}} \leftarrow \mathbf{z} \odot \mathbf{m}$ ▷ Mask tokens
    $\mathcal{L}_{\text{kurt}} \leftarrow \log(1 + \gamma(\hat{\mathbf{z}}))$ ▷ Kurtosis loss
    $\mathcal{L}_{\text{reg}} \leftarrow 1 - \frac{\mathbf{v}\cdot\mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}$ ▷ Regularization loss
    $\mathcal{L} \leftarrow -\lambda \cdot \mathcal{L}_{\text{kurt}} + \mathcal{L}_{\text{reg}}$
    `optimizer.step(`$\mathcal{L}$`)`
    $s \leftarrow s + 1$
  end while
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathbf{R}\}$
  $\mathcal{T} \leftarrow \{i : |z_i - \mu_{\mathbf{z}}| > k \cdot \sigma_{\mathbf{z}}\}$ ▷ High-kurtosis tokens
  $m_i \leftarrow 0 \;\; \forall i \in \mathcal{T}$ ▷ Mask discovered tokens
until $\gamma(\hat{\mathbf{z}}) < \tau$ or $i > n_{\text{iter}}$
return $\mathcal{R}$

C.2 Weight reconstruction analysis

The iterative nature of ROTATE raises two termination questions: (1) when to stop optimizing a single rotation matrix, and (2) how many iterations to run per neuron. For (1), we follow standard practice and terminate when the loss change falls below a threshold ϵ or a maximum step count n_{step} is reached. For (2), rather than attempting to estimate the "polysemanticity degree" of each neuron, we set a fixed iteration budget $n_{\text{iter}} = 50$ and verify empirically that this suffices for high-fidelity reconstruction.

To assess how well the discovered channels collectively reconstruct the original weight vector, we track two metrics across iterations, evaluated on Gemma-2-2B-it. Given channels $\{v_1, \ldots, v_t\}$ discovered after t iterations, we define the residual $r_t = w - \sum_{i=1}^{t}(w \cdot v_i)v_i$ and report: (1) per-channel cosine similarity between each newly discovered channel v_t and w, and (2) cumulative explained norm, defined as $1 - \|r_t\| / \|w\|$.

Figure 8 shows both metrics for 99 randomly sampled neurons per layer and weight type. Early channels capture the dominant directions of w (cosine similarity > 0.9 within $\sim$10 iterations), while later channels contribute smaller but consistent refinements. By iteration 50, the cumulative explained norm approaches 1.0 across all layers and weight types, confirming that 50 iterations suffice to account for nearly all of the original weight vector's norm. The consistent behavior across layers and weight matrices (gate, in, out) indicates that the decomposition is robust to the specific structure of the weight vector.

C.3 Faithfulness and completeness across layers

To assess whether our main results are specific to the layers selected in §5.1, we run ROTATE and the input-side evaluation suite on a sweep spanning early, middle, and late layers of both models.

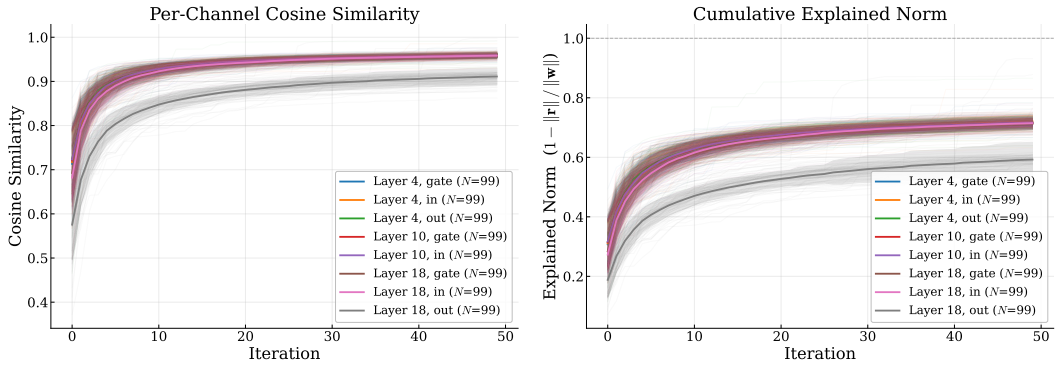

Figure 8: Weight reconstruction analysis on Gemma-2-2B-it. **Left:** Per-channel cosine similarity with the original weight vector w across iterations. **Right:** Cumulative explained norm $(1 - \|r_t\|/\|w\|)$ over iterations. Lines show medians across 99 neurons; shaded regions indicate inter-quartile ranges. Channels collectively reconstruct nearly all of w within 50 iterations across all layers and weight types.

Table 3 reports faithfulness and completeness per layer. Every evaluated layer remains above the random baselines (faithfulness 20–25%, completeness 20%), with the strongest gains concentrated in middle-to-late layers, mirroring the training dynamics of vocabulary kurtosis (Figure 6), which concentrates in the same layers.

	Gemma-2-2b-it		Llama-3.1-8B-Instruct	
Layer	Faithfulness	Completeness	Faithfulness	Completeness
2	46.9	40.7	–	–
4	45.7	55.7	49.7	41.3
8	45.2	43.5	49.3	35.8
18	46.1	59.7	70.9	55.1
22	–	–	57.5	49.3
24	42.9	37.4	–	–

Table 3: Layer sweep of input-side faithfulness and completeness (%). All layers exceed the random baselines (faithfulness 20–25%, completeness 20%), with the strongest gains in middle-to-late layers.

C.4 Channel consistency

Since ROTATE relies on a non-convex optimization procedure with random initialization (Algorithm 1), we evaluate the stability of the algorithm's output as an additional means of validating the method.

Experiment We run ROTATE with 4 different random seeds on the same set of 50 randomly sampled gate neurons from layer 18 of Gemma-2-2B-it. For each neuron, this yields 4 independent sets of discovered channels. To quantify consistency, we measure whether the same channels are recovered across runs. For each pair of runs, we compute the pairwise cosine similarity between all channels from run A and all channels from run B. We then apply greedy matching to find the best one-to-one alignment between the two channel sets. For each matched pair, we compute the Jaccard similarity of their top-k tokens to verify semantic agreement. High similarity across matched pairs indicates that the discovered vocabulary channels are stable features of the weight landscape.

Results We report a mean cosine similarity of 0.9 ± 0.04 and a mean Jaccard similarity of 0.8 ± 0.05 across matched pairs. These high similarity scores demonstrate that ROTATE consistently recovers the same semantic directions regardless of initialization. Figure 9 shows an example for a pair of executions with the matching channels marked. Notably, channels are not always discovered in the same order across runs, as they sometimes appear off-diagonal. This is expected as the random initialization of the Householder vector $\mathbf{h}$ determines which local optimum is found first, while the masking procedure ensures subsequent iterations discover different channels. The consistency of the set of discovered channels, despite varying discovery order, suggests these directions are genuine structures in the weight space rather than artifacts of a particular optimization trajectory.

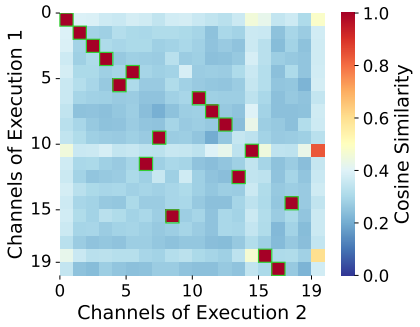

Figure 9: Consistency of ROTATE across different initializations. The heatmap displays a pairwise cosine similarity between vocabulary channels discovered in two separate execution runs (Execution 1 vs. Execution 2) for the same target neuron.

C.5 Avoiding glitch tokens

A practical challenge we encountered is that the optimization frequently converges to "glitch tokens" (Li et al., 2024), which are under-trained token embeddings characterized by extreme norms. Since our objective maximizes kurtosis, it is inherently sensitive to such outliers; the extreme norms of these tokens manifest as high-kurtosis directions that act as degenerate attractors in the optimization landscape. To prevent the algorithm from exploiting these tokenizer artifacts, we initialize the mask $\mathbf{m}$ (Alg. 1, line 3) to exclude known glitch tokens (Land & Bartolo, 2024) and ensure the method focuses on genuine semantic sparsity.

C.6 Ablations

Applying rotations on the same vector To motivate the need for iterative token masking, we compare the standard ROTATE pipeline with token masking between iterations against a variant that performs independent optimization runs with no depletion after each iteration. Meaning neither token masking nor residual subtraction between iterations.

We first demonstrate that without depletion, the optimization landscape contains a single dominant attractor. We run ROTATE on 50 gate, in, and out neurons from Layer 18 of Gemma-2-2B-it, executing 20 independent optimization runs per neuron with different random seeds but no masking between runs. For each run, we record the anchor token (the top token of the vocabulary-projected channel) and the set of top-20 tokens. The mean pairwise Jaccard similarity of top-20 token sets is 0.60, confirming strong semantic agreement even when the exact anchor token differs slightly.

This redundancy directly harms decomposition quality. Figure 10 compares both variants over 20 iterations on the same set of gate neurons. Without depletion, nearly every iteration rediscovers the same dominant direction, yielding a mean cosine similarity of only 0.42 and a mean explained norm of 0.19, indicating that repeated runs contribute almost no additional reconstruction of $\mathbf{w}$. With token masking, subsequent iterations are steered toward novel high-kurtosis directions, achieving a mean cosine similarity of 0.88 and a mean explained norm of 0.78. Consistent patterns hold for $\mathbf{w}_{\text{in}}$ and $\mathbf{w}_{\text{out}}$. These results confirm that depletion is essential: without it, the iterative procedure collapses to a single channel and fails to decompose the neuron.

Applying subtraction instead of masking To prevent the iterative optimization from rediscovering the same semantic directions, ROTATE employs token masking. A standard alternative, common in methods like ICA, is iterative residual subtraction (deflation), where

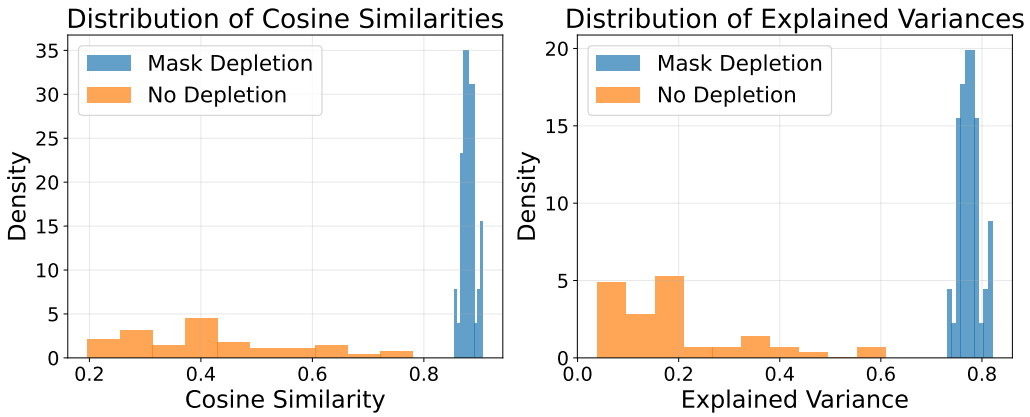

Figure 10: Effect of token masking on iterative decomposition quality. We compare ROTATE with token masking against independent optimization runs with no depletion over 20 iterations on 50 gate neurons from Layer 18, Gemma-2-2B-it. **Left:** Per-channel cosine similarity with $\mathbf{w}$. **Right:** Cumulative explained norm ($1 - \|\mathbf{r}_t\|/\|\mathbf{w}\|$). Without masking, all iterations converge to the same dominant direction, yielding negligible reconstruction progress (mean explained norm 0.19 vs. 0.78).

the projection of the discovered channel is subtracted directly from the weight vector before the next iteration.

As shown in Figure 11, iterative subtraction strictly underperforms token masking in reconstructing the original weight vector. Subtraction captures significantly less of the cumulative explained norm (top row) and achieves lower overall cosine similarity with the original weight (bottom row) across iterations for both W_{gate} and W_{out}. This suggests that geometrically projecting out the channel permanently degrades the weight vector's remaining latent structure, making subsequent feature extraction less effective. Token masking, by contrast, preserves the original geometry of $\mathbf{w}$ while successfully steering the kurtosis objective toward novel semantic directions.

Using more than 1 Householder matrix A single Householder matrix ($k = 1$) is technically a reflection rather than a proper rotation. Composing two Householder matrices ($k = 2$) yields a true rotation. In practice, however, we find that a single reflection is entirely sufficient. As illustrated in Figure 11, the $k = 2$ configuration performs virtually identically to the $k = 1$ baseline across all metrics and weight types, with their curves overlapping almost perfectly. This confirms that a single reflection provides the necessary degrees of freedom to align the basis with high-kurtosis directions, rendering the added complexity and parameterization of multiple Householder matrices unnecessary.

C.7 Hyperparameters selection

Table 4 summarizes the grid search results for our hyperparameter configurations. Hyperparameters were evaluated on a held-out set of 100 neurons per model/layer combination (disjoint from the experimental evaluation set) via grid search over the Cartesian product of: learning rate $\eta \in \{8 \times 10^{-4}, 2 \times 10^{-3}\}$, regularization coefficient $\lambda \in \{0.1, 0.3, 0.5\}$, and standard deviation threshold $\sigma \in \{4.0, 6.0, 8.0\}$.

Because the metrics clustered heavily by the regularization penalty, we report the highest-performing configuration for each λ value. Configurations were ranked by maximizing the harmonic mean of two metrics:

First, *orthogonality score* measures how mathematically distinct the discovered channel directions are from one another. It is defined as 1 minus the mean absolute pairwise cosine

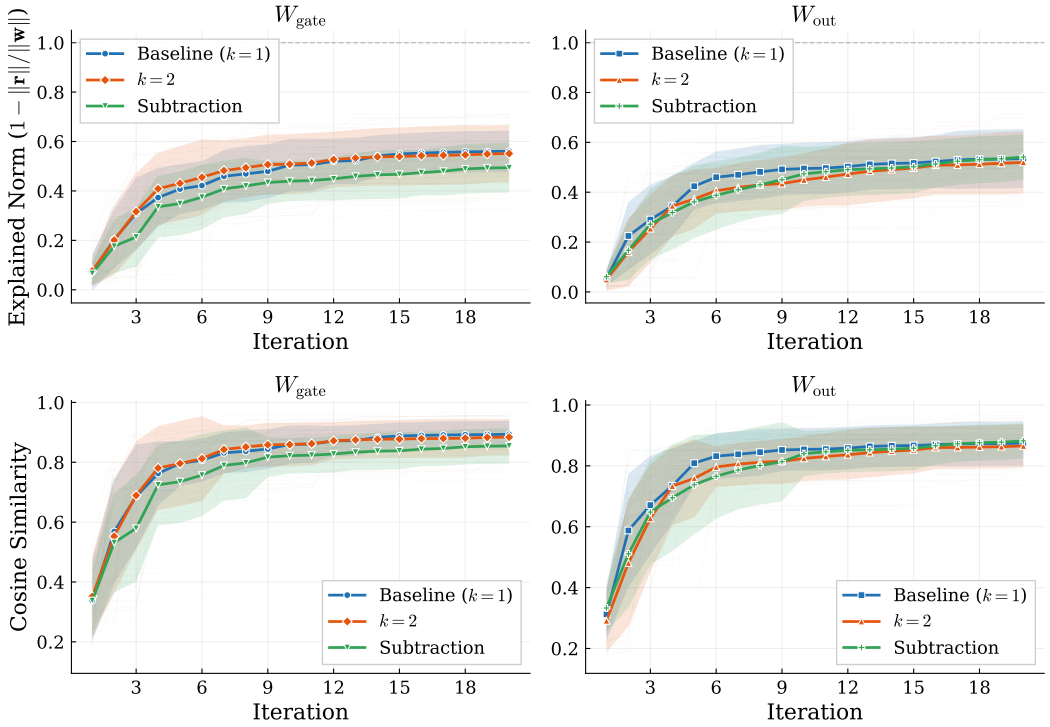

Figure 11: Ablation results evaluating weight reconstruction across optimization iterations for W_{gate} (left) and W_{out} (right). We compare the ROTATE baseline (token masking, $k = 1$ Householder matrix) against two variants: utilizing a proper rotation via two Householder matrices ($k = 2$), and using residual subtraction instead of token masking. **Top:** Cumulative Explained Norm ($1 - \|\mathbf{r}\|/\|\mathbf{w}\|$). **Bottom:** Cosine similarity between the reconstructed vector and the original weight vector. The baseline ($k = 1$) matches the performance of the more complex $k = 2$ parameterization and consistently outperforms residual subtraction.

similarity between all pairs of distinct extracted direction vectors $\mathbf{d}_i$ and $\mathbf{d}_j$:

$$\text{Orthogonality Score} = 1 - \frac{1}{N(N-1)} \sum_{i \neq j} \frac{|\mathbf{d}_i \cdot \mathbf{d}_j|}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|} \tag{7}$$

where N is the total number of channels. Taking the absolute value ensures that both highly correlated and highly anti-correlated directions are penalized.

Second, *explained norm* measures the proportion of the neuron's original magnitude that is captured by the learned channels. It is calculated as 1 minus the relative reconstruction error:

$$\text{Explained Norm} = 1 - \frac{\|\mathbf{w} - \hat{\mathbf{w}}\|}{\|\mathbf{w}\|} \tag{8}$$

where $\mathbf{w}$ is the original neuron weight vector, $\hat{\mathbf{w}}$ is the reconstructed neuron vector, and $\|\mathbf{w} - \hat{\mathbf{w}}\|$ represents the L_2 norm of the reconstruction error (the residual).

The number of optimization steps per channel was fixed at $n_{\text{step}} = 3000$.

C.8 Computational budget

Method efficiency ROTATE operates entirely on model weights and requires no activation data, making its compute cost independent of dataset size. This contrasts sharply with activations-based baseline, which require collecting and processing millions of activation vectors before training can begin.

λ	Best η	Best σ	Explained Norm	Final Orthogonality	Harmonic Mean
0.3	2×10^{-3}	4.0	0.72	0.78	**0.749**
0.5	8×10^{-4}	6.0	0.63	0.87	0.731
0.1	8×10^{-4}	4.0	0.86	0.54	0.663

Table 4: Summary of hyperparameter grid search, reporting the best performing configuration (by harmonic mean) for each kurtosis regularization coefficient (λ). η: learning rate, σ: standard deviation threshold for token masking.

Parallelism and independence Each neuron's optimization is fully independent: the loss and gradient for a neuron depends only on its own rotation matrix and weight vector, with no coupling to other neurons. We exploit this structure by stacking all neurons in a chunk into a single batched tensor of shape $[\text{chunk size}, k, d_{\text{model}}]$ and running gradient descent on all of them in one forward–backward pass, with no interference between neurons. We use chunks of 5,000 neurons. One iteration (extracting one channel per neuron) takes approximately 11 minutes for a chunk of 5,000 neurons on a single H100 GPU.

Hardware and timing All experiments were run on a single NVIDIA H100 GPU. Applying ROTATE to all neurons in one layer (extracting 50 channels per weight vector) takes approximately 3.8 GPU-hours for Gemma-2-2B-it (9,216 neurons per layer) and approximately 6.7 GPU-hours for Llama-3.1-8B-Instruct (14,336 neurons per layer). The 100-neuron experimental sample used for evaluation completes in under 30 minutes per layer.

C.9 Vocabulary channels token-rank analysis

The regularization term in Eq. 2 anchors each channel $\mathbf{v}_i$ to the original weight vector $\mathbf{w}$, achieving high reconstruction similarity (§C.2). This raises a natural concern; the top tokens of a channel projection $\mathbf{v}_i\mathbf{U}$ are merely a re-ordering of the top tokens of the raw neuron projection $\mathbf{wU}$.

To address this, we sample 20 gate neurons from Gemma-2-2B-it (10 each from layers 4 and 18) and, for every channel discovered by ROTATE, compute the rank of each of the channel's top 50 tokens in the raw neuron projection $\mathbf{wU}$. If channels simply re-ranked the neuron's top tokens, these ranks would concentrate at the top of the raw logit list.

We observe that across channels, 97.1% of channel top tokens fall outside the top-200 tokens of the raw projection (rank $\geq$ 200), with a median rank of 122,302 out of $V = 256{,}000$. Vocabulary channels thus surface tokens that direct vocabulary projection of $\mathbf{w}$ never reaches, rather than re-ordering the tokens it already exposes.

C.10 Limitations

ROTATE operates under a deliberate inductive bias: it searches for features that are aligned with the model's vocabulary. A significant body of work has identified functional components that operate in latent subspaces orthogonal to the vocabulary, such as confidence regulation mechanisms (Stolfo et al., 2024) or positional processing features (Voita et al., 2024). Such components fall outside the scope of our decomposition. Nevertheless, our completeness results (§5.4) demonstrate that vocabulary-aligned channels account for a substantial portion of neuron behavior, suggesting that this signal, while not exhaustive, still captures an accessible and significant layer of MLP computation.

In addition, we evaluate two layers per model across two architectures, selected based on alignment to the vocabulary basis. Extending to additional scales, and architectures is a valuable next step.

D Qualitative examples

In this section, we provide example channels obtained by ROTATE (see Table 5) and analyze the interplay between $\mathbf{w}_{\text{gate}}$, $\mathbf{w}_{\text{in}}$, $\mathbf{w}_{\text{out}}$ channels within the gated MLP, illustrating how vocabulary channels getting us closer to better understanding of the mechanisms behind neuron behavior. We examine Neuron 9005 in Layer 18 of Gemma-2-2B-it (Figure 12). This neuron activates positively on technical text involving negation and polarity concepts (e.g., comparison operators in C code, formal identities discussing + and -) and negatively on temporal deferral constructions (e.g., *"it wasn't until 1817"*, *"for many years"*).

Input side: when and why. ROTATE explains this dual behavior through the interaction of gate and value ($\mathbf{w}_{\text{in}}$) channels. On the positive side, $\mathbf{w}_{\text{gate}}$ channel 2 (*"negative, Negative"*) detects contexts where negation or polarity is discussed, while $\mathbf{w}_{\text{in}}$ channel 1 (*"negative, positive"*), a polarity concept signal aligns positively with the input ($\mathbf{w}_{\text{in}} \cdot \mathbf{x} = +1.76$). The product $\sigma(\mathbf{w}_{\text{gate}} \cdot \mathbf{x}) \cdot (\mathbf{w}_{\text{in}} \cdot \mathbf{x})$ is positive, yielding activation $+2.76$. On the negative side, $\mathbf{w}_{\text{gate}}$ channel 0 *"until, Until"*, detects temporal markers, while $\mathbf{w}_{\text{in}}$ channel 6 strongly anti-aligns with these inputs ($\mathbf{w}_{\text{in}} \cdot \mathbf{x} = -2.25$), producing activation -4.53.

Output side: what is promoted. The output-side channels complete the picture by revealing what the neuron writes to the residual stream for each activation sign. Output channels discovered by ROTATE carry both kurtosis (sparsity) and skewness (directionality): positive-skew channels have their semantically meaningful tokens on the positive (promoted) side, while negative-skew channels have them on the negative (suppressed) side. Since a negative neuron activation flips the sign of the output contribution, negative-skew channels effectively have their bottom tokens promoted when the neuron fires negatively.

Concretely, when the neuron fires positively, it promotes polarity vocabulary through output channel 4 (*"negative, positive"*, a polarity concept signal, aligns positively with skewness $= +4.6$), along with code-closing syntax (ch 1, skew $= +8.7$) and dashes (ch 2, skew $= +6.3$). When the neuron fires negatively, the sign flip promotes the bottom tokens of negative-skew channels: negation contractions *"wasn't, didn't, weren't"* (ch 0, skew $= -4.1$), multilingual temporal markers *"until, Till, hasta, jusqu"* (ch 3, skew $= -4.2$), and temporal delay vocabulary *"wait, waiting"* (ch 5, skew $= -4.2$).

This example demonstrates how vocabulary channels provide a much more nuanced and more mechanistic account: the input-side $\mathbf{w}_{\text{gate}} \times \mathbf{w}_{\text{in}}$ decomposition explains *when* and *why* the neuron activates with a particular sign, while the output-side channels, organized by skewness, explain *what* the neuron promotes for each sign. Notably, the output channels reveal that this single neuron implements two coherent but distinct functions depending on activation polarity. All channel are discovered entirely from weights, without any activation data.

E Additional experimental details

E.1 Baselines

SAE-based baselines Following Gur-Arieh et al. (2025b), we disentangle MLP gate neurons using sparse autoencoders (SAEs) as a baseline for comparison with ROTATE. We employ the Gemma Scope and Llama Scope SAEs (Lieberum et al., 2024; He et al., 2024), which are trained on the residual stream at each neuron's respective layer. For each neuron, we take the top $k = 15$ vectors from the SAE's out projection matrix with the highest dot product with said neuron, treating these vectors as the SAE-based counterpart to ROTATE's channels.

TopK-Iter We implement an iterative logit-lens decomposition: project $\mathbf{w}$ through $\mathbf{U}$, take the top $k = 50$ tokens as a "channel", mask them, and repeat. Channel descriptions are produced with the same pipeline as ROTATE channels (§5.1) and evaluated with the identical faithfulness and completeness protocols. We evaluate on 20 neurons (10 per layer) from layers 4 and 18 of Gemma-2-2B-it. TopK-Iter achieves 0.10 faithfulness (vs. ROTATE's 0.46–0.47 on these layers) and 4% completeness, well below the 20% chance level (vs. 55–60%).

Fires Positively (top examples)

`"2)) < (w2)) && (((x1) - (x2)) > -(w1))"`
Code with comparison/negation operators act. = +2.76

`"Operator x - y produces the same result as x + (-y)"`
Formal text on positive/negative polarity act. = +2.74

Fires Negatively (bottom examples)

`"Still, it wasn't until 1817 that the city..."`
Temporal deferral construction act. = −4.53

`"...the utility and effectiveness for many years."`
Temporal duration act. = −3.49

Input Side: $\mathbf{w}_{\text{gate}} \times \mathbf{w}_{\text{in}}$ channel decomposition *(explains **when** and **why** the neuron fires +/−)*

$\mathbf{w}_{\text{gate}}$ ch 2: *"negative, Negative"* ($\sigma = 2.16$)
Detects contexts involving negation/polarity.

$\mathbf{w}_{\text{in}}$ ch 1: *"negative, positive"* ($\mathbf{w}_{\text{in}} \cdot \mathbf{x} = +1.76$)
Polarity concept signal (93% of top examples).
Aligns with input $\Rightarrow \sigma(\cdot) \times (+) > 0$

Predicted: > 0 **True:** $+2.76$

$\mathbf{w}_{\text{gate}}$ ch 0: *"until, Until"* ($\sigma = 4.41$)
Fires on temporal markers (100% of bottom ex.).

$\mathbf{w}_{\text{in}}$ ch 6: *"until, Until"* ($\mathbf{w}_{\text{in}} \cdot \mathbf{x} = -2.25$)
Strongly anti-aligns with temporal contexts.
$\sigma(\cdot) \times (-) < 0$

Predicted: < 0 **True:** -4.53

Output Side: Vocabulary channels with signed skewness *(explains **what** the neuron promotes)*

Positive activation promotes (positive-skew channels):

ch 4 (skew = +4.6): *"negative, positive, Negative"*
Polarity vocabulary—the predicted concept.

ch 1 (skew = +8.7): `']); "]); "));`
Code closing syntax.

ch 2 (skew = +6.3): "--", "—", "—"
Minus sign, dashes and separators.

Negative activation promotes (negative-skew, sign-flipped):

ch 0 (skew = −4.1): *"wasn't, weren't, didn't"*
Negation contractions.

ch 3 (skew = −4.2): *"until, Till, hasta, jusqu"*
Temporal markers (multilingual).

ch 5 (skew = −4.2): *"wait, waiting, waited"*
Temporal waiting/delay.

Figure 12: **Complete mechanistic decomposition of Neuron 9005 (Layer 18, Gemma-2-2B-it) via vocabulary channels. Top:** The neuron activates positively on technical text with negation/polarity concepts and negatively on temporal deferral. **Middle:** ROTATE's input-side $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ channels explain the sign of the activation, the $\mathbf{w}_{\text{gate}}$ detects relevant context, while the $\mathbf{w}_{\text{in}}$ channel's alignment or anti-alignment with the input determines the sign. **Bottom:** Output-side channels, organized by skewness sign, reveal what the neuron writes to the residual stream. Positive activation promotes polarity vocabulary (*"negative"*, *"positive"*); negative activation promotes temporal negation tokens (*"wasn't"*, *"until"*, *"wait"*). All channels are discovered from weights alone.

FastICA As a data-driven counterpart sharing ROTATE's kurtosis-deflation structure, we run FastICA (Hyvärinen & Oja, 2000) (kurtosis contrast, orthogonal deflation) on the top and bottom token embeddings most similar to $\mathbf{w}$, treating the recovered components as channels under the same description and evaluation pipeline (20 neurons sampled from each of layers 4 and 18 of Gemma-2-2B-it). FastICA performs at or below chance, faithfulness 0.25, completeness 0.18, with components converging to a few overlapping directions with ROTATE but missing most of the channels carrying faithfulness and completeness.

Both weight-space alternatives fall far below ROTATE: TopK-Iter reaches a faithfulness of only 0.10 and a completeness of 4%, and FastICA performs at or below chance (faithfulness 0.25, completeness 0.18), despite sharing ROTATE's kurtosis-deflation structure (evaluated on Gemma-2 layers 4 and 18). Neither raw logit-lens inspection nor data-driven kurtosis deflation recovers the structure that ROTATE's weight-anchored rotations do.

Model	Neuron	MLP type	Ch	Top tokens	Description
Gemma-2 -2b-it	(18, 6528)	W_{gate}	0	*ride, Ride, riding, rides, ridden*	Direct riding vocab.
			47	*platform, Platform, platforms*	Platform
			38	*school, School*	Dampens school ctx.
		W_{in}	0	*ride, riding, rides, bike, horseback*	Riding / locomotion
			16	*donkey, donkeys, horse, horses, mule*	Animals / mounts
			22	*gl, Gl, GL*	`gl-` subtoken
		W_{out}	0	*ride, riding, Ride, bike, motorcycle*	Suppresses riding
			1	*mother, Mother, mom, father, parent*	Promotes parenting
			9	*mechanical, Mechanical, mechanism*	Suppresses mechanics
Llama-3.1 -8B-Instruct	(18, 496)	W_{gate}	0	*instruction, instructions, directions*	Instructions
			2	*accept, Accept, acceptance*	Acceptance
			7	*charge, Charge, charges, fee*	Dampens charges/fees
		W_{in}	0	*instructions, directions*	Instructions
			3	*loyalty, loyal, faithful, allegiance*	Loyalty
			4	*control, Control*	Control
		W_{out}	0	*follow, Follow*	Following
			6	*order, orders*	Orders
			7	*submission, submit, obedience*	Submission

Table 5: Selected vocabulary channels for two example neurons, across W_{gate}, W_{in}, and W_{out} weight matrices. Top tokens (up to 5) shown per channel.

E.2 Input-side results

Figure 13 illustrates four representative gate channels of Neuron 9005, showing the top tokens, description, and activating examples for each.

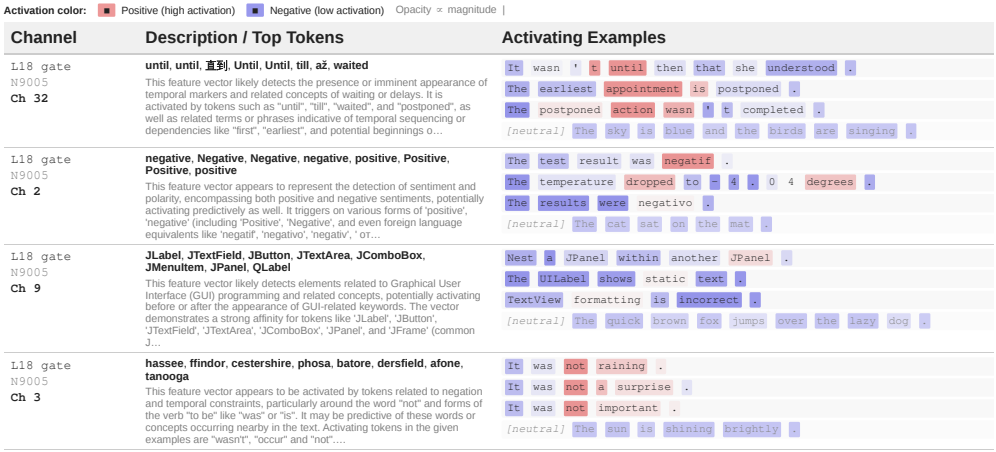

Figure 13: Visualization of four gate channels of Neuron 9005 (Layer 18, Gemma-2-2B-it). Each row shows a channel's top vocabulary tokens, its natural-language description, and three activating examples alongside one neutral example. Token color indicates activation polarity (red: positive, blue: negative) and opacity scales with magnitude. The channels capture distinct concepts: temporal markers, polarity/negation, GUI programming tokens, illustrating the fine-grained, interpretable structure recovered by `ROTATE` from a single neuron's weight vector.

Figure 14 shows the per-channel faithfulness results for the 4 gate channels of Neuron 9005 (Layer 18, Gemma-2-2B-it). For each channel, Gemini-2.0-Flash generates 40 activating and 40 neutral sentences from the channel description; we compare peak neuron activations via a one-sided Welch t-test at $p < 0.05$. The four panels in Figure 14 show representative

passing channels, where activating sentences consistently elicit higher peak activations than neutral ones.

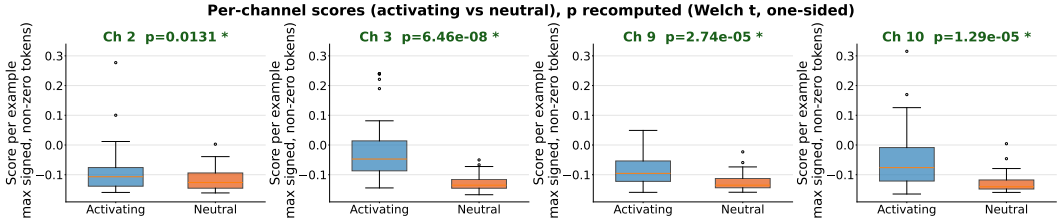

Figure 14: Per-channel faithfulness scores for representative gate channels of Neuron 9005 (Layer 18, Gemma-2-2B-it). Each panel shows the distribution of peak neuron activations on activating (blue) vs. neutral (orange) sentences generated from the channel description. Channels shown all pass the one-sided t-test at $p < 0.05$ (indicated in the title of each panel), confirming that their descriptions reliably distinguish activating from non-activating inputs.

Activating / Neutral Example Generation Prompt Given a channel description, we prompt an LLM to generate synthetic sentences expected to activate the neuron (positive) and sentences that should not (negative), following the protocol described in §5.2. The full prompt is shown in Figure 19.

E.3 Completeness setup

For each gate weight vector we retrieve a random subset of 100 out of its top-1000 activating examples from $\mathcal{D}$ and identify, for each example $\mathbf{x}$, the top channel $\mathbf{v}^* = \arg\max_{\mathbf{v} \in \mathcal{C}} (\mathbf{x} \cdot \mathbf{v})$. We then present an LLM judge (Gemini-3.1-Flash-Lite) with:

1. The activating token context, with the highest-activating token marked `**like this**`.
2. Five candidate descriptions: the description of $\mathbf{v}^*$ (correct) and four distractors drawn uniformly at random from channels of *other* neurons in the same model and layer set.

The judge selects the description it believes best explains why the neuron fired; we record a hit when it selects the correct description.

Example. Below is a sample query for Neuron 9005 (Layer 18, Gemma-2-2B-it), where the neuron fired on the token `**wasn't**`.

Sample Completeness Query — Neuron 9005

Sentence: "It `**wasn't**` what I expected at all."
Candidate descriptions:
1. Riding and locomotion contexts (horses, bikes, vehicles).
2. Polarity/negation constructions: contractions like *w*asn't, didn't, can't. **[correct]**
3. Instruction-following and obedience vocabulary.
4. Technical programming and software development tokens.
5. Temporal markers indicating future scheduling.

Judge response: "The sentence contains 'wasn't', a negation contraction. Description 2 best matches."

The four distractor descriptions are sampled from random neurons in Gemma Layer 18. In this example the judge selects Description 2, the correct vocabulary channel.

E.4 Patchscopes setup

We use the Patchscopes framework (Ghandeharioun et al., 2024) to decode semantic content encoded in a neuron's output weight vector $\mathbf{w}_{\text{out}}$. We construct the few-shot prompt

$$\underbrace{\texttt{cat} \rightarrow \texttt{cat;}\ \texttt{135} \rightarrow \texttt{135;}\ \texttt{hello} \rightarrow \texttt{hello;}}_{\text{few-shot context}} \quad ?$$

where the ? probe token's residual-stream representation (at the input to block 0) is overwritten with the scaled weight vector $\alpha\,\mathbf{w}_{\text{out}}$ before the forward pass continues. The few-shot context biases the model to "read" the semantic content of the injected vector rather than predicting from syntactic context alone.

Why scaling by α is necessary. Token embeddings in Gemma-2-2B-it have ℓ_2 norm on the order of $\|\mathbf{e}_t\| \approx 100$–$150$, whereas MLP output weight vectors have norm $\|\mathbf{w}_{\text{out}}\| \approx 0.5$–$2$. Injecting the raw weight vector ($\alpha = 1$) therefore places the probe far outside the distribution of token embeddings, yielding near-degenerate generations. Multiplying by α rescales the probe into the normal embedding range:

$$\mathbf{p}_\alpha = \alpha\,\mathbf{w}_{\text{out}}.$$

We sweep $\alpha \in \{-400, -350, \ldots, 350\}$ (step 50). Setting $\alpha > 0$ amplifies the semantic content of $\mathbf{w}_{\text{out}}$; setting $\alpha < 0$ probes its *semantic opposite* by flipping the injected direction, which for a dual-polarity neuron surfaces the other polarity cluster.

Channel ablation. To test the causal role of a specific channel $\mathbf{v}$, we ablate it from $\mathbf{w}_{\text{out}}$ before injecting:

$$\mathbf{w}_{\text{ablated}} = \mathbf{w}_{\text{out}} - \frac{\mathbf{w}_{\text{out}} \cdot \mathbf{v}}{\|\mathbf{w}_{\text{out}}\|^2}\,\mathbf{v},$$

where $\mathbf{v}$ is the channel vector (not unit-normalised). The weight $(\mathbf{w}_{\text{out}} \cdot \mathbf{v})/\|\mathbf{w}_{\text{out}}\|^2$ measures how much of $\mathbf{w}_{\text{out}}$'s length is contributed by $\mathbf{v}$. We then inject $\alpha\,\mathbf{w}_{\text{ablated}}$ and compare the decoded output to the baseline injection $\alpha\,\mathbf{w}_{\text{out}}$ at $\alpha = 400$.

Decoding parameters. We run 20 independent sampling passes for each alpha value of the baseline and 10 for each alpha value ablated variant (temperature $= 0.9$, up to 8 new tokens per pass). All generated tokens are pooled into a single multi-set per condition.

Metric. Let $T_{\mathbf{v}}$ be the top-50 vocabulary-projection tokens of channel $\mathbf{v}$. Define the concept-token fraction for a weight vector $\mathbf{w}$ as

$$f(\mathbf{w}) = \frac{\left|\{t \in \text{pool}(\mathbf{w}) : t \in T_{\mathbf{v}}\}\right|}{|\,\text{pool}(\mathbf{w})|}.$$

The relative change when channel $\mathbf{v}$ is ablated is

$$\Delta = \frac{f(\mathbf{w}_{\text{ablated}}) - f(\mathbf{w}_{\text{out}})}{f(\mathbf{w}_{\text{out}})} \in (-1, 1), \qquad \mathbf{w}_{\text{ablated}} = \mathbf{w}_{\text{out}} - (\mathbf{w}_{\text{out}} \cdot \mathbf{v})\,\mathbf{v}.$$

Self-channel ablation monitors the fraction of $T_{\mathbf{v}}$ tokens when $\mathbf{v}$ itself is ablated; *cross-channel ablation* monitors the same fraction when a different channel $\mathbf{v}' \neq \mathbf{v}$ is ablated instead. A faithful, non-redundant channel should produce $\Delta_{\text{self}} \approx -1$ and $\Delta_{\text{cross}} \approx 0$.

Example. For out-channel 0 of Neuron 9005 (top tokens: *wasn't, weren't, didn't, can't, isn't*), self-ablation reduces the fraction of polarity tokens from ≈18% to ≈2% ($\Delta \approx -89\%$), while cross-ablation of an unrelated channel leaves it near 18% ($\Delta \approx +15\%$).

E.5 LLM judge validation

Two evaluation tasks in this paper rely on LLM judges: completeness (§5.4), judged by Gemini-3.1-Flash-Lite, and head-to-head description comparison (§6), judged by Gemini-3-Flash. We use different judges as the completeness task is simpler and requires substantially more LLM calls, making a lightweight model preferable. To assess whether these LLM judges are reliable substitutes for human annotators (NLP graduate students), we apply the Alternative Annotator Test (Calderon et al., 2025), which tests whether an LLM can statistically replace a human annotator within an annotation group. For each task, three annotators independently annotated 50 instances following the same protocols as the LLM judge. For the head-to-head task, description order was randomized and annotators were

blind to method identity. We set $\varepsilon = 0.15$, which is suited for skilled annotators, and a $p{-}value = 0.05$.

On the completeness task , Gemini-3.1-Flash-Lite achieves $\bar{\rho}_f = 0.89$ (vs. $\bar{\rho}_h = 0.81$ for humans), with $\omega = 2/3$. On the head-to-head task , Gemini-3-Flash achieves $\bar{\rho}_f = 0.897$ vs. $\bar{\rho}_h = 0.885$, with $\omega = 3/3$. Both tasks pass the $\omega \geq 0.5$ threshold, confirming that the LLM judges can reliably substitute for human annotation in these comparative evaluation settings.

F Additional Details on Neuron Description Generation

F.1 Variant Selection via Pairwise Evaluation

Vocab-channel aggregation strategies We experimented with four strategies for aggregating the 25 gate and 25 $\mathbf{w}_{\text{in}}$ channel descriptions into a single per-polarity neuron description. The variants differ in (a) which gate channels are included and (b) how $\mathbf{w}_{\text{in}}$ channels are filtered by skewness polarity. Table 6 summarizes the four strategies.

Variant	$\mathbf{w}_{\text{gate}}$ channels	$\mathbf{w}_{\text{in}}$ channels
All gate, all in	all	all
Positive-skew gate, all in	positive-skew only	all
All gate, positive-skew in	all	positive-skew only
All gate, negative-skew in	all	negative-skew only

Table 6: Four aggregation strategies for `ROTATE` neuron descriptions. The last two variants separate positive and negative activation regimes by filtering $\mathbf{w}_{\text{in}}$ channels according to the sign of their vocabulary-projection skewness, while retaining all $\mathbf{w}_{\text{gate}}$ channels.

MaxAct baseline variants We evaluated three versions of the MaxAct+VocabProj baseline, differing in what information is provided to the LLM: **v1**: top-20 activating examples only (one combined description); **v2** (selected): top-20 examples concatenated with the top-50 vocabulary tokens from the $\mathbf{w}_{\text{in}}$ and $\mathbf{w}_{\text{gate}}$ vector projections, producing polarity-split descriptions; **v3**: same as v2 but with $\mathbf{w}_{\text{in}}$ and $\mathbf{w}_{\text{gate}}$ vocabulary projections described separately before synthesis.

Stage 1 evaluation To select the best variant within each method, we ran pairwise LLM-judged comparisons (Gemini-2.0-Flash) across all variants, separately for positive- and negative-polarity activation contexts. We used 20 randomly sampled neurons from Llama-3.1-8B-Instruct, with 50 examples per neuron sampled from the top-1000 Pile activations. Position bias was controlled by running each comparison twice with swapped description order and declaring a winner only when both orderings agree. Table 7 reports the win rates.

This section details the full prompting pipeline used in §6.

F.2 Channel-level description

Each of the 25 $\mathbf{w}_{\text{gate}}$ and 25 $\mathbf{w}_{\text{in}}$ channels is independently described by prompting an LLM with the channel's top-50 vocabulary tokens and up to 5 top-activating examples. The full prompt is shown in Figure 18.

F.3 Neuron-level synthesis (polarity-split)

The individual channel descriptions are then synthesized into a single neuron description, separately for positive and negative activations. $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ channel descriptions are provided together, organized by role. The full prompt is shown in Figure 15.

Method	Polarity	Variant	Win rate
ROTATE	positive	all_gate_split_positive	**78.3%**
		all_gate_all_in	35.0%
		positive_gate_all_in	31.7%
ROTATE	negative	all_gate_split_negative	**57.5%**
		all_gate_all_in	37.5%
MaxAct+VocabProj	positive	v2	**67.5%**
		v1	57.5%
		v3	25.0%
MaxAct+VocabProj	negative	v2	**47.1%**
		v1	61.8%
		v3	40.6%

Table 7: Stage 1 within-method variant win rates on 20 neurons from Llama-3.1-8B-Instruct. Bold denotes the selected variant for each method and polarity. For ROTATE, we select all_gate_split_positive (positive) and all_gate_split_negative (negative). For MaxAct+VocabProj, we select v2 for as it enriches the activation-based evidence with vocabulary-projection tokens from both $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$, providing the baseline with the strongest available signal and ensuring the most competitive comparison against ROTATE.

Baseline: MaxAct+VocabProj description For the MaxAct+VocabProj baseline, we prompt the LLM with 20 top-activating examples and the top/bottom-50 vocabulary tokens from the $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ weight vector projections. The full prompt is shown in Figure 16.

Head-to-head pairwise evaluation For the LLM-judged pairwise comparison described in §6, each comparison is run twice with swapped description order; a winner is declared only when both orderings agree. The full judge prompt is shown in Figure 17.

F.4 Head-to-head examples

Table 8 presents selected head-to-head comparisons between ROTATE's unified neuron descriptions and those produced by the MaxAct++ and MaxAct+VocabProj baselines. For each neuron, we show the descriptions generated by all three methods alongside a representative activating example from the Pile positive split. The final column indicates whether the LLM judge preferred the ROTATE description for that example. These cases illustrate how ROTATE's vocabulary-grounded decomposition often yields more specific and faithful descriptions, particularly for neurons encoding structured or syntactic patterns that activation-based methods tend to summarize in overly generic terms.

G Prompts used in experiments

Channel description Each channel is described by prompting an LLM with the channel's top-50 vocabulary tokens and up to 5 top-activating examples. The full prompt is shown in Figure 18.

Activating / neutral example generation prompt Given a channel description, we prompt an LLM to generate synthetic sentences expected to activate the neuron (positive) and sentences that should not (negative), following the protocol described in §5.2. The full prompt is shown in Figure 19.

Completeness LLM judge prompt The 5-way channel matching prompt used for the completeness evaluation is shown in Figure 20.

Layer, Neuron	Activating example	ROTATE description	Baseline description	Win/Loss
L22, N6946	/// Get the host name associated with the entry. template <class Allocator> std*::*basic_string, std::char_traits<char>, Allocator> host_name(const Allocator& alloc **Pile top [100-500]**	This neuron activates on contexts related to sleep, rest, and altered states of consciousness (dreaming, falling asleep), alongside concepts of returning or restarting, often involving function words (*to, of, you*) and morphological elements. Additionally, it responds to notions of bursting/failure, central locations/functions, and suspension/hanging, and **code snippets related to filtering operations on arrays**.	This neuron activates on words related to sleep, sleeping, snoring, and waking up, as well as general personal pronouns and common function words like "to", "or", and "of", possibly reflecting awareness of narrative context involving sleep. **[MaxAct+VocabProj]**	Win
L22, N1939	"thumbnail", "file", "fanart", "streamdetails"], "***player***id": 1], "id": "VideoGetItem" Check this out} **Pile top [0-100]**	This neuron activates in contexts blending organizational systems, financial elements, and technical details, particularly those involving data processing and structured information. This includes: pipelines and routing of data, archives and architecture, financial assets and payments, macro/micro scale comparisons, lists/catalogs, letters/alphabets, notes/records, and measurements of volume. It is also sensitive to names and identifiers, particularly those containing the letter sequence 'ee'	This neuron activates on code snippets, particularly related to the VLC media player library (libVLC) or JSON-**RPC calls for media players** (like XBMC), often involving player control methods. It also activates on articles, 'the' and 'to' **[MaxAct+VocabProj]**	Loss
L12, N496	We just cruised on her to the Panama Canal last week! The Maitre'De in* **the*** Posh Dining Room Goran Gorigjewski is awesome!! **Pile top [0-100]**	This neuron activates positively in contexts involving the definite **article 'the' alongside varied semantic themes** including: workplace interactions; self-reference; code overrides; strength/resilience; sending/transmission; philosophical concepts/proper nouns; authentication ('login'); geographical locations/cardinal directions; physical actions; and potentially female names. This suggests an emphasis on contextually defined entities within narrative or technical contexts	proper nouns; context indicating inquiry or explanation **[MaxAct++]**	Win
L18, N2241	The English prose ***poem*** is a verse form that is usually unrhymed and written in the... **FineWeb top [0-100]**	This neuron strongly activates on code snippets, configurations, and technical documentation, often featuring specific numerical identifiers, compound words, and elements related to authorship or provenance. It also demonstrates sensitivity to partial words and specific syllables ('an', 'on', 'ol', 'ug', 'ac') and common suffixes. Addition-related terms, Slavic language fragments, and spoiler/coupon contexts can also trigger activation.	references to **poetic forms**, styles, or innovation **[MaxAct++]**	Loss

Table 8: Example wins and losses of ROTATE in head-to-head comparisons against MaxAct++ and MaxAct+VocabProj.

Polarity-Split Neuron Description Synthesis Prompt

You are analyzing a neuron in a language model. Below are descriptions of individual channels that correspond to {polarity} activations of the neuron. Try to make this compressed as you can, but still touch on the diverse features of the neuron. A description should be no more than 50 words.
Layer: {layer_idx} Neuron: {neuron_idx} Polarity: {polarity}
$\mathbf{w_{gate}}$ **Channel Descriptions** (control what activates the neuron):
{gate_descriptions}
$\mathbf{w_{in}}$ **Channel Descriptions** (determine the neuron's activation — {polarity} activation):
{in_descriptions}
Your task is to synthesize these channel descriptions into a single, coherent description of what causes {polarity} activations of this neuron.
Create a unified description that:
1. Identifies the common semantic or syntactic themes across channels
2. Explains what inputs activate this neuron
3. Notes any patterns in token appearance vs prediction
4. Is specific enough to be useful but general enough to capture the neuron's overall function

Avoid vague descriptions: Do NOT use generic, uninformative descriptions like "diverse set of linguistic and semantic features" or "various textual patterns". Be SPECIFIC about what causes {polarity} activations. If the channels are truly diverse, list the 2–3 most prominent specific patterns rather than using vague umbrella terms.
Please return your answer in JSON format.

Figure 15: Polarity-split neuron description synthesis prompt (§6). $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ channel descriptions are provided separately; the LLM produces a unified description of at most 50 words. Used with Gemini-2.0-Flash.

MaxAct+VocabProj Baseline Description Prompt

You are analyzing a neuron in a language model.
Layer: {`layer_idx`} Neuron: {`neuron_idx`} Analysis Type: {`polarity_upper`} ACTIVATIONS
Below are the {`polarity_description`} activating examples for this neuron (gate $\times$ in activation). The activating token in each example is marked with `**asterisks**`. The number in brackets `[X.XX]` is the activation value.
{**`polarity_upper`**} **Activating Examples:** {`examples`}
Additionally, here is the LogitLens analysis showing which tokens the neuron's gate and in vectors project to in vocabulary space:
$\mathbf{w}_{\text{gate}}$ **Vector Projection:**
— Top tokens (most similar in vocabulary space): {`gate_top_tokens`}
— Bottom tokens (least similar): {`gate_bottom_tokens`}
$\mathbf{w}_{\text{in}}$ **Vector Projection:**
— Top tokens: {`in_top_tokens`}
— Bottom tokens: {`in_bottom_tokens`}
Your task is to analyze these examples and LogitLens projections to describe what causes {`polarity`} activations. The description should be no more than 50 words.
Consider:
1. What semantic or syntactic patterns appear in these {`polarity`}-activation examples?
2. How do the LogitLens tokens relate to {`polarity`} activation patterns?
3. Is there a coherent theme?

Avoid vague descriptions: Do NOT use generic, uninformative descriptions like "diverse set of linguistic and semantic features" or "various textual patterns". Be SPECIFIC. If the examples are diverse, list the 2–3 most prominent specific patterns rather than using vague umbrella terms.
Please return your answer in JSON format.

Figure 16: MaxAct+VocabProj baseline description prompt (§6). Combines 20 top-activating examples with LogitLens vocabulary projections of the $\mathbf{w}_{\text{gate}}$ and $\mathbf{w}_{\text{in}}$ vectors. Used with Gemini-2.0-Flash.

Head-to-head pairwise evaluation

You are comparing two neuron descriptions to determine which more accurately explains an activating pattern.
Below is a sequence of tokens that highly activates a specific neuron. Tokens with high activation values are marked with their activation in brackets.
Activating Sequence: {`formatted_tokens`}
Description A: {`description_a`}
Description B: {`description_b`}
Task: Determine which description more accurately identifies the specific pattern that causes this neuron to activate on the highlighted tokens.
Important guidelines:
- Focus on ACCURACY, not on level of detail or length. A short, precise description can be better than a long, vague one.
- Do NOT prefer a description just because it is longer or more detailed.
- A description may cover multiple themes. It should win if at least one of its themes correctly explains the highlighted example. Do not penalize a description for covering themes beyond what appears in this example.
- Choose "TIE" if both descriptions capture the activation pattern equally well, even if one is more detailed.

Choices: "A" if Description A more accurately identifies the activation pattern; "B" if Description B does; "TIE" if both are equally accurate (or equally inaccurate).
Response Format (JSON only):
```
{{"winner": "A" or "B" or "TIE",
"reasoning": "<one sentence explanation>"}}
```

Figure 17: Head-to-head pairwise evaluation prompt (§6). Each comparison is run twice with swapped order; a winner is declared only when both orderings agree. Used with Gemini-3-Flash.

Channel Description Prompt

I am analyzing a channel (component) of a neuron in a language model.
Associated tokens (sorted by logit value): {tokens_str}
Additionally, here are real text examples where this channel activates strongly: {examples_section}
Task:
1. Identify the common semantic or syntactic theme among these tokens and examples.
2. Provide a short description of what this channel likely represents or detects.
3. The description should be specific but capture the general concept.

Please return your answer in JSON format.

Figure 18: Prompt used to describe a single vocabulary channel. Each channel is described independently before synthesis into a neuron-level description. Used with Gemini-2.0-Flash.

Activating / neutral example generation prompt

I'm going to give you explanations and interpretations of features from LLMs. You must take in each explanation, and generate {num_positive} sentences for which you think the feature will have a high activation, and {num_negative} for which they'll have a low activation.
For the high activation examples:
- Make sure to choose ones that will cause a high activation with high confidence
- You don't have to include all groups, just make examples that you're confident will have high activation
- Make the sentences both include words from the explanation AND represent the concept
- Try to use specific examples and make them literal interpretations of the explanation, without trying to generalize

For the low activation examples:
- These should have nothing to do with the interpretation
- They should be orthogonal and completely unrelated to the feature

Output Format: You must output strictly valid JSON with the following structure:
{{"max_activation": ["example 1", "example 2", ...],
"min_activation": ["example 1", "example 2", ...]}}
Explanation: {explanation}

Figure 19: Prompt used to generate activating and neutral examples for the input-side faithfulness evaluation (§5.2). Default: 40 positive + 40 negative examples. Used with Gemini-2.0-Flash.

5-Way Channel Matching Prompt (Completeness)

You are going to be given a sentence, and five descriptions of different components in a language model. One of these descriptions describes a component that was highly activated when processing the sentence, while the other four descriptions are unrelated. The most highly activated token in the sentence is marked with double asterisks (**like this**). Your task is to identify this description. Please respond with a short line describing your reasoning, and then the second line should contain only the number (1–5) corresponding to the correct description. Please respond exactly in this format. Even if you are unsure, make your best guess.
Sentence: "{activating_string}"
Descriptions: {chan_descs}

Figure 20: 5-way channel matching prompt used for the completeness evaluation (§5.4). The LLM judge (Gemini-3.1-Flash-Lite) selects which of five candidate descriptions best matches the activating input.