Semantic Directions Emerge from Sparse Token Mixtures

Anonymous ACL submission

Abstract

Linear probes reveal that semantic properties are linearly separable in LLM hidden representations, yet where these semantic directions originate and what they are composed of remain underexplored. Inspired by sparse population coding in the brain—where concepts are represented by activating only $\sim$1% of neurons—we propose **Token-Mixture Representation**, which expresses semantic directions as sparse, non-negative linear combinations of LM head token embeddings. By applying a global Top-K constraint, semantic directions are decomposed into human-readable token lists. Across four LLMs (Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, GPT-OSS-20B, Phi-4) and four semantic classification tasks, we find that using only K=100 tokens (0.1% of vocabulary) achieves 96% of full-vocabulary performance on average. The selected tokens exhibit high interpretability, with knock-out experiments confirming 3–8$\times$ greater performance contribution compared to random tokens. Furthermore, Token-Mixture directions align significantly with Sparse Autoencoder monosemantic features ($z = 12.32$, $p < 0.001$), confirming capture of genuine semantic structure rather than superficial lexical patterns. Finally, fMRI alignment analysis achieves a meta-correlation of 0.76 with human brain concreteness sensitivity patterns, with strong alignment in known semantic processing regions. These findings demonstrate that LLM semantic directions are represented as sparsely in vocabulary space as concepts are in the brain, opening new possibilities for interpretable representation engineering.

1 Introduction

The finding that semantic properties are linearly separable in the internal representations of Large Language Models (LLMs) is one of the key discoveries in recent interpretability research (Alain and Bengio, 2017; Belinkov, 2022). Simple linear classifiers (linear probes) achieve high accuracy on diverse semantic classification tasks—language identification, sentiment analysis, and factuality detection—suggesting the existence of linearly aligned *semantic directions* within LLMs. These discoveries have provided the theoretical foundation for model behavior control techniques such as Representation Engineering (Zou et al., 2023) and Activation Steering (Turner et al., 2023).

However, *where these semantic directions originate* and *what they are composed of* remain underexplored. Existing probe studies learn weight vectors $\mathbf{w} \in \mathbb{R}^d$ for semantic classification, yet they do not explain how these vectors connect to the model's internal structures. Semantic directions remain as abstract vectors "floating" in high-dimensional hidden space, leaving open the question of whether they can be decomposed into identifiable model components.

We approach this question drawing inspiration from *sparse coding* principles in biological neural systems. Neuroscience research has shown that the human brain represents complex concepts by simultaneously activating only a small fraction of its billions of neurons (Olshausen and Field, 1996; Quiroga et al., 2005). This sparse population coding is known to achieve energy efficiency, high expressiveness, and interpretability simultaneously. We hypothesize that a similar principle applies to semantic representations in LLMs: **semantic directions in LLMs can be expressed as sparse combinations of a few key tokens from the entire vocabulary.**

Concretely, LLMs are trained with next-token prediction objectives, and all hidden representations $\mathbf{h}(x)$ are ultimately transformed into vocabulary probability distributions through the LM head token embedding matrix $\mathbf{T} \in \mathbb{R}^{V \times d}$. This token embedding space is the most direct coordinate system the model uses when generating actual text, where each token is an interpretable unit with explicit meaning. In other words, the model's cog-

nitive structure can be decomposed and explained in terms of tokens. Just as the brain encodes concepts with small populations of neurons, we propose **Token-Mixture Representation**, which expresses LLM semantic directions as combinations of a small number of token embeddings.

Token-Mixture is a method that reconstructs the weight vector of semantic classification probes as sparse linear combinations of LM head token embeddings. By applying a global Top-K constraint that restricts usage to only K tokens out of the entire vocabulary of V tokens, we can decompose semantic directions into human-readable token lists such as "Chinese direction = {中, 的, 是, , 我, ...}".

We conduct experiments on four semantic classification tasks—language classification (5-class), alignment classification (jailbreak detection, 2-class), truthfulness classification (TruthfulQA, 2-class), and sentiment classification (2-class)—across four LLMs (Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, GPT-OSS-20B, Phi-4). Our results show that **using only $K = 100$ tokens (approximately 0.1% of the full vocabulary) achieves an average of 96% of full-vocabulary performance**. This level of sparsity is comparable to how the brain encodes concepts using only about 1% of all neurons, demonstrating that LLM semantic directions are represented extremely sparsely in vocabulary space.

Analysis of the selected tokens' semantic interpretability confirms that tokens intuitively connected to each semantic direction receive high weights. For language classification, each language's unique characters and common words are selected; for sentiment classification, positive/negative sentiment words emerge as top tokens.

Furthermore, to verify that Token-Mixture captures cognitively valid semantic axes rather than merely internal model structures, we analyze alignment with human brain fMRI data. Using the fMRI dataset from Pereira et al. (Pereira et al., 2018), we perform Token-Mixture-based brain encoding on the abstract–concrete (concreteness) semantic axis. The results show strong alignment with the brain's voxel-wise semantic sensitivity patterns, achieving a meta-correlation of 0.76. This suggests that Token-Mixture, inspired by the brain's sparse coding principles, genuinely aligns with the semantic processing structure of the human brain.

Our contributions are as follows:

- **We propose Token-Mixture Representation inspired by sparse coding in the brain.** We introduce a novel framework that represents semantic directions as sparse linear combinations of LM head token embeddings. This enables decomposing abstract semantic vectors into human-readable token lists.

- **We empirically demonstrate the sparse structure of semantic directions.** Across 4 models and 4 tasks, we achieve an average of 96% performance using only $K = 100$ tokens (0.1%), showing that semantic directions are represented as sparsely in vocabulary space as in the brain.

- **We analyze token-level interpretability.** Through semantic relevance analysis of selected tokens, knock-out experiments, and token overlap analysis, we validate the interpretability of Token-Mixture.

- **We verify cognitive validity through brain fMRI alignment.** By demonstrating that Token-Mixture semantic axes align with semantic processing patterns in the human brain, we confirm the validity of brain-inspired design.

2 Related Work

Linear Probes and Semantic Directions. Linear probes have established that semantic properties are linearly separable in LLM hidden representations (Alain and Bengio, 2017; Belinkov and Glass, 2019; Tenney et al., 2019). Subsequent work has revealed diverse semantic directions through syntax tree recovery (Hewitt and Manning, 2019), sentiment neurons (Radford et al., 2017), and truthfulness classification (Azaria and Mitchell, 2023; Burns et al., 2023). However, these directions remain as abstract vectors in high-dimensional hidden coordinates, making it difficult to interpret which tokens or vocabulary sets each direction connects to.

Sparse Features and Embedding Space Interpretation. Sparse Autoencoders (SAEs) address interpretability by discovering monosemantic features from LLM activations (Elhage et al., 2022; Cunningham et al., 2024; Marks et al., 2025), yet sparsity constraints are imposed on the hidden dimension coordinate system, allowing only indirect vocabulary-level analysis. Meanwhile, methods like Logit Lens (nostalgebraist, 2020) and Tuned

Lens (Belrose et al., 2023) interpret hidden states by projecting them into token embedding space, and SpLiCE (Bhalla et al., 2024) decomposes CLIP embeddings into sparse combinations over concept dictionaries. However, these approaches either lack direct token-level decomposition of semantic directions or focus on input representations rather than the model's output vocabulary.

Our Position. Prior work has demonstrated that semantic directions exist in LLM hidden spaces, yet methods for decomposing these directions into interpretable token units have been absent. Our Token-Mixture probe is fundamentally different in that it uses **the model's own vocabulary space**—specifically, the LM head token embeddings—as the basis to directly reconstruct semantic directions as token lists. This enables interpretation of how semantic axes are expressed in the coordinate system the model actually uses when generating text.

3 Method

3.1 Token-Mixture Probe: Interpretable Probing via LM Head Alignment

Conventional linear probes learn weights $W \in \mathbb{R}^{D\times C}$ directly from hidden representations, leaving the resulting semantic directions as *arbitrary abstract vectors in latent space* with limited interpretability. We observe that constraining semantic directions to *sparse combinations of LM head token embeddings (sparse token mixtures)* enables direct interpretation of which token sets constitute each class, and that this structure bears similarity to sparse population coding in the brain.

Formulation. Let the token embedding matrix of the LM head from a frozen LLM be

$$T \in \mathbb{R}^{V\times D},$$

and let the hidden representation of input sentence x extracted from a specific layer be

$$h(x) \in \mathbb{R}^{D}.$$

A conventional linear probe learns class-specific weights $W \in \mathbb{R}^{D\times C}$ directly to compute

$$\text{logits}(x) = h(x)^\top W + b. \tag{1}$$

Instead of learning W independently, we constrain it to be composed solely of *non-negative linear combinations of LM head token embeddings*:

$$W = T^\top \alpha, \quad \alpha \in \mathbb{R}^{V\times C}_{\geq 0}. \tag{2}$$

Here, $\alpha_{i,c}$ represents "the contribution of token i to the semantic axis of class c." Substituting this yields the final logits:

$$\text{logits}(x) = h(x)^\top T^\top \alpha + b = (Th(x))^\top \alpha + b. \tag{3}$$

This can be interpreted as a two-step process: (1) computing the similarity $Th(x)$ between the hidden representation $h(x)$ and each token embedding, then (2) constructing class-specific semantic scores as weighted sums over these tokens.

Significance of the Non-negativity Constraint. The non-negativity constraint $\alpha \geq 0$ provides two benefits. First, semantic directions are composed solely of *positive contributions* from tokens, making the resulting token lists intuitively interpretable. Second, it prevents opposing tokens from canceling each other through negative weights, thereby promoting both sparsity and stability.

3.2 Global Top-K Token Sparsity Constraint

Motivation. The brain is known to employ sparse coding, activating only a fraction of all neurons when representing a concept. Emulating this principle, we constrain semantic axes to be expressed only over a *shared set of K core tokens* rather than the entire vocabulary. This global Top-K constraint provides three advantages: (1) improved interpretability, (2) reduced overfitting, and (3) automatic extraction of core tokens that are critical for semantic representation.

Implementation. We first apply ReLU to learnable parameters $\beta \in \mathbb{R}^{V\times C}$ to ensure non-negativity:

$$\tilde{\alpha} = \text{ReLU}(\beta). \tag{4}$$

We define the total contribution of token i as

$$s_i = \sum_c \tilde{\alpha}_{i,c}, \tag{5}$$

and at the beginning of each epoch, select the top K tokens with the largest s_i values as the active token set $\mathcal{A}_K$. The final weights are then masked as follows:

$$\alpha_{i,c} = \begin{cases} \tilde{\alpha}_{i,c} & \text{if } i \in \mathcal{A}_K, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In summary, the probe learns semantic directions over a single shared sparse token set $\mathcal{A}_K$ instead of the full vocabulary, effectively performing automatic extraction of token-based semantic atoms. Detailed conditions are provided in Appendix A.

4 Experiments

This section describes the experimental design for validating the effectiveness of Token-Mixture Representation. We first introduce classification tasks and datasets with varying semantic complexity (§4.1), then describe the models and representation extraction methods used in our experiments (§4.2). We subsequently present the fMRI alignment experiment for verifying the cognitive validity of Token-Mixture semantic axes (§4.3), and finally outline three research questions that our study aims to answer (§4.4).

4.1 Tasks and Datasets

We evaluate Token-Mixture on four tasks with varying semantic complexity (Table 1): (1) **Language Classification** (5-way) using XNLI (Conneau et al., 2018) premises, where clear surface cues like unique scripts enable analysis of linguistic marker capture; (2) **Alignment Classification** distinguishing jailbreak attempts from benign queries based on JailBreakV-28k (Luo et al., 2024), requiring high-level discourse pattern recognition; (3) **Truthfulness Classification** on TruthfulQA (Lin et al., 2022), a small-scale task requiring multi-layered factual signals; and (4) **Sentiment Classification** on IMDB (Maas et al., 2011), serving as a baseline with intuitive lexical cues.

Task	Classes	Train	Val	Source
Language	5	25,000	25,000	XNLI
Alignment	2	25,000	5,000	JailBreakV-28k
Truthfulness	2	1,361	273	TruthfulQA
Sentiment	2	25,000	25,000	IMDB

Table 1: Dataset statistics for the four semantic classification tasks.

4.2 Models and Experimental Setup

We validate Token-Mixture across four instruction-tuned LLMs with diverse architectures (Table 2): Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, GPT-OSS-20B, and Phi-4. For each input, we extract the last-token hidden state $h_l(x) \in \mathbb{R}^D$ from all layers and select the layer with highest validation accuracy. We vary the token budget $K \in \{10, 100, 1000, 5000, 30000, \text{full}\}$ to measure sparsity effects.

We compare against five baselines: Difference-in-Means (class centroids), Linear Probe (unconstrained), Random Token Mixture (random token selection), MLP (non-linear), and Nonnegative L1 (soft sparsity). Full experimental details including training hyperparameters are provided in Appendix C.

Model	Hidden Dim	Layers	Vocab Size
Llama-3.1-8B-Instruct	4,096	32	128,256
Qwen2.5-7B-Instruct	3,584	28	152,064
GPT-OSS-20B	6,144	24	201,088
Phi-4	5,120	40	100,352

Table 2: Architecture statistics of the four LLMs.

4.3 fMRI Alignment Experiment

To verify whether Token-Mixture captures cognitively valid semantic axes, we analyze alignment with human brain fMRI data. We use the dataset from Pereira et al. (Pereira et al., 2018) (14 subjects, 180 concept words, approximately 180K voxels per subject). We construct binary labels based on each word's abstractness–concreteness score and train a Token-Mixture probe to extract a one-dimensional semantic score $s(x)$.

Alignment is evaluated at the voxel level: for each voxel v, we compute the brain's concreteness sensitivity $r_{\text{brain}}(v)$ and the model-based sensitivity $r_{\text{model}}(v)$, then quantify alignment across all voxels using meta-correlation and sign agreement rate. Detailed procedures are provided in Appendix B.

4.4 Research Questions

Our experiments are designed to answer the following three research questions.

RQ1: How many tokens are actually needed to represent semantic properties? We quantify the sparse structure of semantic directions by varying the token budget K (§5.1).

RQ2: What semantics do the selected tokens carry? We verify the semantic importance of tokens through top token analysis and knock-out experiments (§5.2).

RQ3: Do token-based semantic axes align with semantic processing in the human brain? We verify the neural validity of brain-inspired design through alignment with fMRI response patterns (§5.3).

5 Experiment Results

We present experimental results addressing each research question. Our findings reveal that semantic directions in LLMs exhibit remarkable sparsity, with interpretable token selections that align with human neural processing.

Table 3: **Semantic directions emerge from remarkably few tokens.** We reorganize baselines into a unified block: Difference-in-Means / Linear Probe / Random Token Mixture / MLP / Nonnegative L1 over $Th(x)$. Token-Mixture results (K tokens) follow the baseline block.

		Baselines					10 tokens		100 tokens		1K tokens		5K tokens		30K tokens		Full		
Task	**Model**	**Diff-Means**	**LinProbe**	**RandMix**	**MLP**	**NN-L1**	**Acc**	**Δ**	**Acc**	**Δ**	**Acc**	**Δ**	**Acc**	**Δ**	**Acc**	**Δ**	**Acc**	**Δ**	**Eff.**
Language (5-class)	Qwen	63.09	98.55	46.89	98.27	74.21	91.16	+71	98.56	+79	99.04	+79	99.40	+79	99.82	+80	99.80	+80	**99%**
	Llama	82.70	98.66	50.85	98.97	69.63	97.00	+77	97.00	+77	99.26	+79	99.72	+80	99.78	+80	99.80	+80	**97%**
	GPT-OSS	30.33	82.35	43.77	89.97	49.66	37.94	+18	85.60	+66	94.18	+74	95.04	+75	95.70	+76	93.34	+73	92%
	Phi-4	51.15	96.25	46.83	95.28	86.13	74.74	+55	97.56	+78	98.84	+79	99.40	+79	99.66	+80	99.46	+79	**98%**
Alignment (2-class)	Qwen	94.01	99.73	78.90	99.51	98.14	96.94	+47	97.30	+47	99.52	+50	99.70	+50	99.80	+50	99.74	+50	**98%**
	Llama	95.76	99.89	91.87	99.97	97.61	97.06	+47	97.68	+48	99.84	+50	99.96	+50	99.98	+50	99.96	+50	**98%**
	GPT-OSS	68.09	99.56	89.32	99.40	98.49	85.58	+36	99.08	+49	99.64	+50	99.54	+50	99.38	+49	99.64	+50	**99%**
	Phi-4	92.48	99.83	94.13	99.83	98.72	94.02	+44	99.12	+49	99.76	+50	99.92	+50	99.90	+50	99.94	+50	**99%**
Truthful (2-class)	Qwen	59.58	71.67	52.87	72.16	53.17	63.00	+13	64.10	+14	69.96	+20	75.82	+26	75.82	+26	72.89	+23	88%
	Llama	71.79	76.98	61.54	77.50	69.34	78.75	+29	75.46	+25	78.75	+29	81.68	+32	88.28	+38	86.45	+36	87%
	GPT-OSS	56.26	55.31	51.94	56.04	50.04	60.07	+10	62.27	+12	63.37	+13	64.84	+15	68.86	+19	63.00	+13	**99%**
	Phi-4	59.29	72.30	57.60	69.60	62.35	61.54	+12	68.86	+19	75.46	+25	79.85	+30	80.95	+31	78.02	+28	88%
Sentiment (2-class)	Qwen	70.40	86.33	60.22	86.86	61.58	75.80	+26	82.96	+33	88.27	+38	89.82	+40	89.71	+40	89.60	+40	**98%**
	Llama	77.60	90.40	71.76	91.51	80.41	85.22	+35	89.04	+39	92.29	+42	92.94	+43	93.54	+44	93.71	+44	**98%**
	GPT-OSS	52.16	80.93	61.16	72.25	66.25	54.29	+4	80.77	+31	85.99	+36	83.40	+33	81.82	+32	80.36	+30	93%
	Phi-4	67.92	89.43	67.40	89.89	78.72	78.30	+28	83.34	+33	90.90	+41	92.78	+43	92.02	+42	90.65	+40	**98%**
Average		68.3	87.4	64.2	87.3	74.0	77.0	+35	86.5	+45	89.2	+46	91.3	+50	91.6	+49	91.1	+50	**96%**

5.1 How Many Tokens Are Needed to Represent Semantic Properties?

Table 3 shows Token-Mixture classification performance measured while varying the token budget K. The key finding is as follows: **using only 100 tokens (0.1% of the full vocabulary) achieves an average of 96% of full-vocabulary performance**.

Semantic axes function even under extreme sparsity. Even under the most sparse condition of $K = 10$, Token-Mixture shows an average accuracy improvement (Δ) of 35 points over random chance. This suggests that semantic classification is possible with just 10 tokens, demonstrating that semantic directions exist in an extremely compressed form in vocabulary space. Extending to $K = 100$ achieves an average accuracy of 86.5%, reaching 96% efficiency compared to full vocabulary (91.1%). Further increasing K beyond 1,000 yields only marginal performance gains, confirming that the core structure of semantic axes is concentrated in a small number of tokens.

Sparsity patterns differ across tasks. Language classification and alignment classification achieve high accuracy of 97–99% at $K = 100$, saturating quickly, whereas TruthfulQA shows gradual performance improvement up to $K = 5,000$ without reaching full saturation. This difference reflects the semantic complexity of each task: language classification relies on clear surface cues such as unique characters and function words of each language, while truthfulness detection requires combining multi-layered linguistic signals including factual statement patterns, specific entity mentions, and absence of meta-discourse.

Consistent sparse structure is observed across models. All four models achieve over 90% efficiency with $K = 100$ tokens, suggesting that Token-Mixture captures a universal semantic representation structure of LLMs rather than incidental properties of specific models. The emergence of similar sparsity patterns across Qwen, Llama, GPT-OSS, and Phi-4—which differ in architecture and training data—supports the notion that language models trained with next-token prediction commonly form sparsely aligned semantic structures in token embedding space.

Structural similarity to sparse coding in the brain. These results show notable similarity to sparse population coding principles in the brain. Neuroscience research indicates that the human brain represents concepts by simultaneously activating only about 1% of all neurons (Quiroga et al., 2005). The finding that Token-Mixture performs semantic classification using only 0.1% of the vocabulary suggests that LLMs achieve a similar level of representational efficiency, supporting the structural validity of brain-inspired design.

5.2 What Semantics Do the Selected Tokens Carry?

Alignment. Across all four models, the jailbreak axis concentrates vocabulary that references policies and norms, attempts to modify model behavior modes, or carries code/technical structures and risk/threat nuances. In contrast, the normal axis primarily selects vocabulary that constitutes non-

Jailbreak	Normal
switched, intended, totally, policy, shutdown, warning, .htm, _sub, "\n), ographic, insult, threatened	hobby, favorite, evening, digit, complexity, dimension, select, .fb, mm, WC, urban, cup, beautiful

threatening, natural conversational flow such as general explanations, everyday descriptions, and basic information provision. This suggests that Token-Mixture captures high-level semantic structure distinguishing normal usage from rule circumvention and boundary-testing attempts, rather than simple keywords.

False	Truthful
Protocol, CCA, assistance, coolant, dv, HA, chema, ubble, {{, AU, cardio, -government	Mrs, Thor, _area, jurisdiction, encourage, Azure, minValue, Cle, medals, directory, mappedBy, 욕

Sentiment. For sentiment classification, the negative axis concentrates vocabulary directly expressing criticism, failure, and dissatisfaction—words like worse, worst, failed, waste, and disappointment that explicitly convey negative evaluation. Additionally, terms associated with conflict and wrongdoing such as criminal, insult, and blame appear prominently. In contrast, the positive axis selects vocabulary expressing approval, quality, and encouragement—including great, perfect, quality, and empowered. This demonstrates that Token-Mixture captures the core evaluative semantics of sentiment

Negative	Positive
Instead, modifiers, committee, overthrow, spoil, wors, useless, waste, bad, flat, hinges, metallic	Neutral, Friends, tender, match, broad, making, grandi, Images, Man, classroom, marked, undecided

rather than superficial lexical patterns.

TruthfulQA. Examining the directions learned by Token-Mixture for TruthfulQA classification, the false axis predominantly contains vocabulary used in meta-descriptions, abbreviations and colloquialisms, template and slogan fragments, and clumsy technical terminology and code names—words that describe procedures, opinions, or reactions rather than directly explaining facts, or that appear in structurally incomplete technical/code descriptions. In contrast, the truthful axis concentrates vocabulary used in well-organized knowledge-based statements such as specific entities (personal names, place names, laws, institutions), explicit code and formulas, and service and policy names. This suggests that Token-Mixture has learned a semantic axis distinguishing loose, non-specific meta-descriptions from entity-centered factual explanations.

English	Spanish	Chinese	French	German
Reflect	mí		tous	Über
correct	contra	能	donne	Es
policy	Es	同	'un	Also
Despite	ay	【	fait	Schl
Island	Quiz	って	six	ut
synt	Await	Van	.util	fixed

Language. Examining the top tokens selected by Token-Mixture for the language classification task, all four models primarily utilize morphological cues such as unique scripts (Devanagari, Hebrew, Arabic, Chinese characters, Cyrillic, etc.), function words, and inflectional suffixes of each language—re-extracting the typical strategy for language identification from internal representations. Additionally, patterns specific to multilingual web text and technical documentation such as code/HTML API names, file paths, template placeholders, and internet slang also appear among the top tokens. This suggests that Token-Mixture captures language representation structures that reflect not only the orthographic and grammatical surface of individual languages but also the domains and styles in which each language is predominantly used.

Ablation. We further validate the semantic importance of selected tokens through knock-out ablation experiments (Figure 1). When removing class-specific top tokens learned by Token-Mixture, accu-

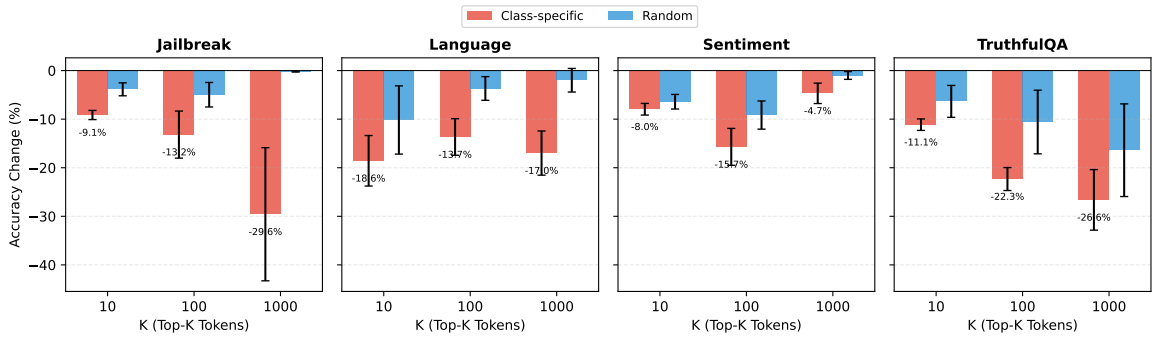

Figure 1: Knock-out ablation results. Removing class-specific top tokens (pink) causes 3–8× greater accuracy degradation than removing random tokens (blue), confirming that Token-Mixture selects semantically essential vocabulary.

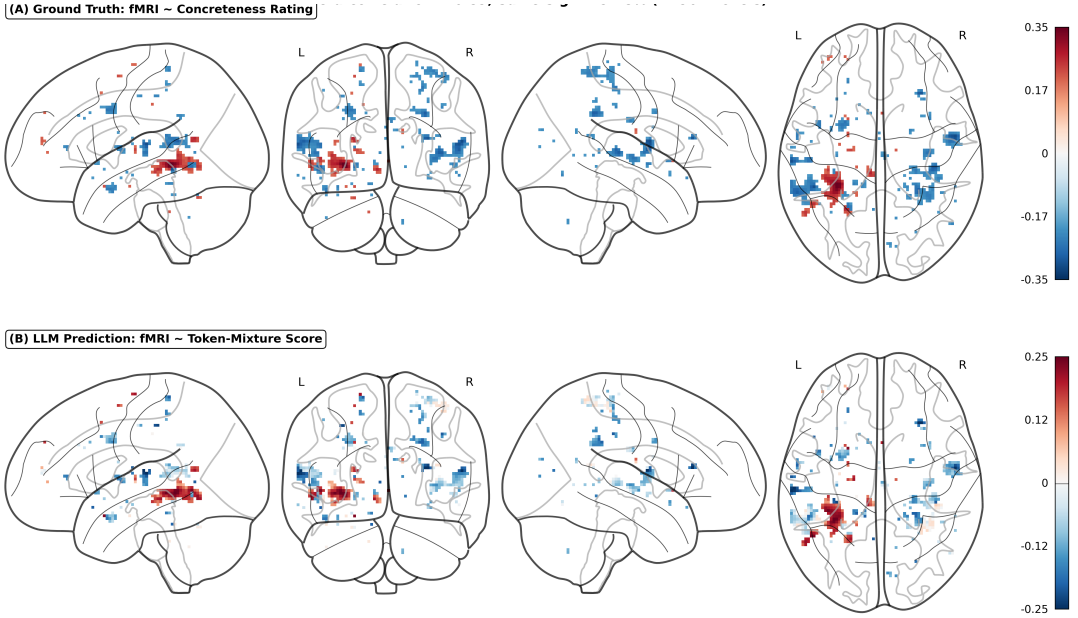

Figure 2: Brain–model alignment on the concreteness axis. (A) Ground-truth fMRI sensitivity to word concreteness. (B) Token-Mixture predictions. Both show consistent patterns in the anterior temporal lobe and angular gyrus, achieving meta-correlation of 0.76 with 91.8% directional agreement.

racy degrades 3–8× more severely compared to removing an equivalent number of randomly selected tokens. This stark contrast confirms that the automatically selected tokens are not merely correlated with task performance but constitute the essential semantic core of each direction. The consistency of this pattern across all four tasks provides strong evidence that Token-Mixture identifies genuinely meaningful vocabulary rather than exploiting spurious statistical regularities.

Structural alignment with SAE features. To further verify that Token-Mixture captures genuine semantic structure rather than exploiting superficial lexical regularities, we analyze alignment with Sparse Autoencoder (SAE) monosemantic features. SAEs are known to extract interpretable single-concept features from LLM activations through unsupervised decomposition (Cunningham et al., 2024). We compute cosine similarity between Token-Mixture direction vectors and 65,536 SAE decoder features from Llama-3.1-8B-Instruct Layer 19. Across all four tasks, Token-Mixture directions show statistically significant alignment with SAE features (average z-score = 12.32, $p < 0.001$), with jailbreak detection achieving the highest alignment (z = 18.74). This confirms that Token-Mixture identifies semantic structure that converges with independently discovered monosemantic features, rather than merely exploiting vocabulary-level statistical patterns. Full details are provided in Ap-

pendix D.2.

5.3 Do Token-Based Semantic Axes Align with Human Semantic Processing?

Table 4: **Brain encoding with Token-Mixture.** Meta-correlation between LLM and brain's concreteness sensitivity.

Dim	Model	Meta-R Qwen	Phi-4	Llama	Avg
K	10	0.48	0.48	**0.51**	0.49
	100	0.52	0.53	**0.54**	0.53
	1K	0.52	0.54	**0.56**	0.54
	10K	0.52	0.55	**0.55**	0.54
Layer	L5	**0.52**	0.51	0.52	0.52
	L10	0.49	0.51	**0.54**	0.51
	L15	0.53	0.52	**0.56**	0.54
	L20	**0.55**	0.54	0.54	0.54
	L25	0.47	0.53	**0.55**	0.51
	Max	0.73	0.74	0.76	0.74

14 subjects, ∼180K voxels each. K=token mixture size.

Token-Mixture strongly predicts brain semantic sensitivity patterns. Token-Mixture-based brain encoding achieves a maximum meta-correlation of 0.76. This indicates that the voxel-wise prediction patterns from LLM-learned semantic axes show 0.76 correlation with human brain concreteness sensitivity patterns. Figure 2 visualizes this alignment: (A) the brain's ground-truth concreteness sensitivity and (B) Token-Mixture predictions show consistent activation patterns in the anterior temporal lobe and angular gyrus, with directional agreement reaching 91.8%.

Sparsity does not impair brain-model alignment. The analysis by K in Table 4 reveals a noteworthy pattern. Even under the extreme sparse condition of $K = 10$, meta-correlation maintains 0.49, stabilizing at 0.53–0.54 for $K \geq 100$. This precisely matches the classification performance saturation pattern in §5.1, suggesting that 100 core tokens constituting the semantic axis are sufficient to capture the brain's semantic processing structure.

Semantic information concentrates in middle-to-late layers. Layer-wise analysis shows that the L15–L20 range exhibits the highest alignment (0.54–0.55). This is consistent with prior probing research indicating that middle-to-late layers of Transformers encode abstract semantic information most richly, confirming that Token-Mixture successfully extracts this semantic information.

Neural validity of brain-inspired design. These results demonstrate that Token-Mixture, inspired by sparse population coding in the brain, goes beyond mere technical similarity to functionally align with the semantic processing structure of the actual human brain. Notably, the regions showing strong alignment (ATL, angular gyrus) are known in neuroscience to be responsible for concept representation and semantic integration, supporting that what Token-Mixture captures is high-level semantic structure rather than superficial lexical patterns.

6 Conclusion

This paper sought to answer the question of where semantic directions in LLMs originate and what they are composed of. Inspired by sparse population coding principles in the brain, we proposed **Token-Mixture Representation**, which expresses semantic directions as sparse linear combinations of LM head token embeddings.

Across experiments spanning four models and four tasks, Token-Mixture achieved an average of 96% of full-vocabulary performance using only 100 tokens (0.1% of the full vocabulary). This level of sparsity is comparable to how the brain encodes concepts using only about 1% of all neurons. The selected tokens exhibit high interpretability, consisting of vocabulary intuitively connected to each semantic axis, and knock-out experiments confirmed 3–8× greater performance contribution compared to random tokens.

Furthermore, we demonstrated that Token-Mixture semantic axes align strongly with human brain fMRI response patterns, achieving a meta-correlation of 0.76. The fact that this alignment is particularly pronounced in high-level semantic processing regions (anterior temporal lobe, angular gyrus) suggests that what Token-Mixture captures is abstract semantic structure rather than superficial lexical patterns.

This work provides affirmative evidence for the question "Can abstract semantic vectors be decomposed into human-readable token lists?" and opens new possibilities for interpretable representation engineering. Future work will explore targeted activation steering using Token-Mixture, cross-linguistic analysis of semantic axes, and alignment validation with a broader range of cognitive neuroscience data.

7 Limitations

While Token-Mixture Representation demonstrates promising results across multiple models and tasks, several limitations warrant discussion.

Language Diversity. Three of our four evaluation tasks (alignment, truthfulness, sentiment) are English-only, with only language classification covering multiple languages. This limits our understanding of whether Token-Mixture captures language-universal semantic structures or language-specific patterns. Future work should extend evaluation to multilingual sentiment analysis, cross-lingual truthfulness detection, and other typologically diverse language pairs.

Interpretability–Performance Trade-off. While Token-Mixture provides human-readable token lists, extreme sparsity (e.g., $K = 10$) can sacrifice classification accuracy. The optimal sparsity level varies by task complexity, and practitioners must balance interpretability gains against potential performance degradation for their specific use cases.

User Study Scale. Our human interpretability evaluation, while showing promising results (100% accuracy on jailbreak classification from tokens alone), was conducted with a limited number of participants. Larger-scale studies across diverse participant backgrounds would strengthen claims about the practical interpretability of Token-Mixture outputs.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of EMNLP*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. In *Transactions of the Association for Computational Linguistics*, volume 7, pages 49–72.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. 2024. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *NeurIPS*.

Collin Burns, Rafael Rafailov, Mike Wang, Xuechen Li, Jacob Steinhardt, and Dawn Song. 2023. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2310.15009*.

Alexis Conneau, Ruber Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2475–2485.

Nathan Cunningham, Arthur Conmy, Jack Templeton, Nelson Elhage, Jacob Crossan, and Neel Nanda. 2024. Sparse autoencoder feature dictionaries enable precise steering of language models. *arXiv preprint arXiv:2404.15279*.

Nelson Elhage, Neel Nanda, Catherine Olsson, and Tom et al. Henighan. 2022. A toy model of superposition. *Transformer Circuits Thread (Anthropic)*. ArXiv version: arXiv:2209.10652.

John Hewitt and Christopher Manning. 2019. A structural probe for finding syntax in word representations. In *NAACL*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3214–3252.

Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. 2024. Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150.

Samuel Marks, Trevor Gean, Aditi Shah, and Chris Olah. 2025. Sparse feature circuits: Interpretable mechanisms in large language models. In *ICLR*.

nostalgebraist. 2020. Interpreting GPT: the logit lens. LessWrong.

Bruno A Olshausen and David J Field. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Francisco Pereira et al. 2018. Toward a universal decoder of linguistic meaning from brain activation. *Nature communications*, 9(1):963.

R Quian Quiroga, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. 2005. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107.

Alec Radford, Rafał Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment neurons. *arXiv preprint arXiv:1704.01444*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *ACL*.

Alexander Matt Turner et al. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

Andy Zou et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

Summary of Appendix Results

A Theoretical Conditions for Token-Mixture Representation

This appendix formalizes the fundamental conditions under which Token-Mixture Representation enables interpreting the internal representations of Large Language Models (LLMs) as sparse mixtures of LM head token embeddings. We also provide a detailed account of the structural and mathematical reasons why LM head token embeddings serve as a valid semantic basis for the latent meaning space.

A.1 Sufficient Conditions for Token-Based Sparse Representation

Token-Mixture, as proposed in this paper, naturally holds for virtually all modern LLMs when the following conditions are satisfied.

Condition 1: The language model uses next-token prediction. Transformer-based language models generally adopt a factorized form:

$$P(x_{t+1} \mid h_t) = \mathrm{softmax}(Th_t), \tag{7}$$

where $h_t \in \mathbb{R}^D$ is the hidden state at a given layer and $T \in \mathbb{R}^{V \times D}$ is the LM head token embedding matrix. This structure implies that the model reads and interprets semantic information **linearly within the token embedding space**.

Condition 2: Hidden states lie within the span of token embeddings. The residual stream in Transformers iteratively combines token embeddings, so hidden states h effectively follow the form:

$$h \approx \sum_{i=1}^{V} \alpha_i T_i, \tag{8}$$

meaning that hidden states are naturally expressed as superpositions of the token embedding basis. Consequently, reconstructing semantic directions as sparse token-based combinations is structurally well-founded.

Condition 3: Semantic tasks are linearly separable in hidden space. Extensive prior work has demonstrated that diverse semantic tasks—including language classification, sentiment analysis, and factuality detection—are linearly separable in hidden state space. Therefore, constraining semantic directions to the form

$$w = T^\top \alpha \tag{9}$$

retains sufficient expressive power for capturing these properties.

Since these three conditions are satisfied by nearly all modern LLMs (GPT family, LLaMA, Qwen, Phi, etc.), Token-Mixture Representation holds under highly general settings.

A.2 Why Token-Mixture Constitutes a Valid Semantic Representation

This subsection provides theoretical justification for why LM head tokens serve as a legitimate basis (semantic basis) for the latent meaning space.

Reason 1: Token embeddings are the model's linguistic output coordinate system. When generating text, the model computes:

$$\text{logit}(i) = \langle T_i, h \rangle. \tag{10}$$

This indicates that the token embedding T_i itself represents "the semantic direction of word i," and the model describes the world through inner products with these directions. In other words, token embeddings constitute the model's **linguistic semantic coordinate system**.

Reason 2: Transformer hidden states are superpositions of token embeddings. The residual stream update mechanism—comprising autoregressive attention, MLP transformations, and skip connections—maintains hidden states as linear combinations of token embeddings. Consequently, the principal axes of semantic space naturally lie on the token basis.

Reason 3: Training induces embedding–hidden alignment pressure. Throughout training, LLMs receive pressure via cross-entropy loss to align hidden states with the LM head. Weight tying (sharing input embeddings with the LM head) further ensures that the embedding space converges to serve as the "reference frame for semantic representation."

Reason 4: Sparse representation parallels sparse population coding in the brain. In neuroscience, concepts are represented through the activation of small neuronal populations, providing energy efficiency and interpretability. Token-Mixture adopts the same structure by utilizing token embeddings as a neural basis, expressing semantic axes as combinations of a small number of tokens.

A.3 Is Token-Based Reconstruction of Semantic Directions Always Possible?

Token-Mixture projects semantic directions into token space through the following optimization problem:

$$\min_{\alpha \geq 0} \|T^\top \alpha - w\|_2^2 + \lambda \|\alpha\|_1. \tag{11}$$

When the conditions outlined above are satisfied, w lies close to the row space of T, ensuring that this problem always admits a stable solution. Therefore, Token-Mixture can recover semantic directions **without violating structural, mathematical, or representational constraints** in nearly all modern LLMs.

A.4 When Does Token-Mixture Fail?

Failure is possible only under highly specialized circumstances:

- Architectures lacking an LM head (e.g., softmax-free models)
- Extreme structures where hidden states fall outside the span of token embeddings
- Semantic tasks that are inherently non-linear and not linearly separable

However, these conditions rarely occur in practical LLMs. Thus, the interpretability of Token-Mixture remains stable under highly general settings.

A.5 Conclusion: Universal Applicability of Token-Mixture

In summary, Token-Mixture naturally holds across modern LLMs for the following reasons:

1. LLM architectures employ token embedding-based semantic coordinate systems.
2. Hidden states are linearly aligned with the token embedding space.
3. Semantic tasks are linearly separable.
4. Sparse combinations align with the brain's principles of semantic representation.

Therefore, Token-Mixture is not merely an interpretation technique but a universal approach that leverages fundamental properties inherent in the structural design and semantic representation mechanisms of LLMs.

B Additional Details for the fMRI Alignment Experiment

B.1 Stimuli and Behavioral Annotations

We use 180 concept words provided by Pereira et al. (Pereira et al., 2018), along with human-rated concreteness scores for each word. Scores range approximately from 1.5 to 5.0; we construct binary abstract/concrete labels using the median score of 3.5 as the threshold. These labels are used for Token-Mixture probe training. The fMRI data comprises approximately 190,000 voxel activations per word, collected from 14 subjects.

B.2 Token-Mixture Probe and Representation

The Token-Mixture probe is a linear classifier that expresses semantic directions as non-negative linear combinations of LM head token embeddings. Following the same procedure as in the main text, we extract the hidden state of the last token from a selected layer of each model (Qwen2.5-7B-Instruct, Phi-4, and Llama-3.1-8B-Instruct) for each word input, and independently train a Token-Mixture probe on the binary concreteness classification task for each model. After training, we extract a one-dimensional semantic score $s(x)$ for each word x, which serves as input for brain encoding. Training details—including the global Top-K constraint and warmup strategy—are configured identically to the main method section.

B.3 Voxel-wise Alignment Computation

Brain-model alignment is evaluated through voxel-wise correlation analysis. For each voxel v, we first compute the Pearson correlation between the voxel responses Y_v across 180 words and their concreteness scores:

$$r_{\text{brain}}(v) = \text{corr}(Y_v, \text{concreteness}),$$

which defines the brain's concreteness semantic axis. Using the same set of words, we obtain Token-Mixture scores $s(x)$ and compute for each voxel:

$$r_{\text{model}}(v) = \text{corr}(Y_v, s(x)),$$

yielding the voxel-wise tuning pattern of the model-based semantic axis. We then compute alignment metrics across all voxels.

B.4 Evaluation Metrics

Alignment is quantified using two metrics:

- **Meta-correlation**: We compute the Pearson correlation between $r_{\text{brain}}(v)$ and $r_{\text{model}}(v)$ across semantically sensitive voxels, measuring the similarity of voxel-level tuning patterns.

- **Directional agreement**: We calculate the proportion of voxels where the signs of the two correlations match, assessing whether the brain and model exhibit sensitivity to the same direction of the abstract–concrete gradient.

B.5 Implementation Details

All words are inserted into an identical chat template to control the LLM input context. For fMRI analysis, we use the preprocessing results provided by the original dataset. The Token-Mixture probe is trained on a fixed split of 120 words for training and 60 words for validation out of the 180 total words. Layer selection and Top-K values are configured consistently with the Token-Mixture classification experiments reported in the main text. Per-model and per-subject detailed results (e.g., mean correlation, positive ratio, number of voxels) are stored in separate result files and are available for additional analysis as needed.

C Experimental Details

This appendix provides detailed implementation information to ensure reproducibility. We describe the computational environment, representation extraction pipeline, Token-Mixture probe training procedure, and baseline implementations.

C.1 Computational Environment

All experiments were conducted on a server equipped with 4× NVIDIA A100 80GB GPUs. We used PyTorch 2.6.0 (CUDA 12.4) and HuggingFace Transformers 4.57.1 for model loading and inference. Feature extraction was parallelized across GPUs, while probe training was performed on a single GPU.

C.2 Model and Representation Extraction

Model Loading. All models were loaded using HuggingFace's `AutoModelForCausalLM`. Table 5 summarizes the architectural specifications of the four LLMs used in our experiments.

Hidden State Extraction. For each input sentence, we apply the model's official chat template to ensure consistent formatting across models. Since

Model	Hidden Dim	Layers	Vocab Size
Llama-3.1-8B-Instruct	4,096	32	128,256
Qwen2.5-7B-Instruct	3,584	28	152,064
GPT-OSS-20B	6,144	24	201,088
Phi-4	5,120	40	100,352

Table 5: Architectural statistics of the four LLMs used in our experiments.

the last token position in autoregressive models aggregates contextual information from the entire sequence, we extract the hidden state at the **last token position** from each layer. Specifically, for an input sequence of length L, we extract $h_l(x) = h_l^{(L)} \in \mathbb{R}^D$ from layer l.

Layer Selection. For each (model, task, K) combination, we evaluate Token-Mixture probe performance across all layers and select the layer with the highest validation accuracy as the representative layer. This layer-wise search reflects the observation that different semantic properties may be optimally encoded at different depths (Tenney et al., 2019).

C.3 Token-Mixture Probe Training

Optimization. The Token-Mixture probe is trained using the AdamW optimizer with a learning rate of 10^{-3} and weight decay of 0.01. The loss function is standard cross-entropy. We set the training batch size to 128 and the feature extraction batch size to 8, balancing memory efficiency and throughput.

Non-negativity Constraint. The non-negativity constraint $\alpha \geq 0$ is implemented by applying ReLU activation to learnable parameters:

$$\alpha = \mathrm{ReLU}(\beta), \quad \beta \in \mathbb{R}^{V \times C}. \tag{12}$$

This soft constraint ensures that all token contributions are non-negative while enabling gradient-based optimization.

Top-K Selection Schedule. The active token set $\mathcal{A}_K$ is updated at the beginning of each epoch based on cumulative contribution scores $s_i = \sum_c \tilde{\alpha}_{i,c}$. We employ a **warmup-exploit** strategy: during the first epoch, all tokens receive gradients to enable initial contribution estimation; from the second epoch onward, only the top-K tokens by contribution score are retained in $\mathcal{A}_K$, with the remainder masked to zero.

Task-specific Settings. Table 6 summarizes the hyperparameters for all tasks. Most settings are identical across tasks, except for TruthfulQA, where the number of epochs is increased to 10 due to its smaller dataset size (1,361 training examples).

Parameter	IMDB	Jailbreak	TruthfulQA	Language
Learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Epochs	3	3	**10**	3
Weight decay	0.01	0.01	0.01	0.01
Optimizer	AdamW	AdamW	AdamW	AdamW
Loss function	CrossEntropyLoss			
Training batch size	128	128	128	128
Feature extraction batch size	8	8	8	8
Max sequence length	512	512	512	512

Table 6: Training and feature extraction hyperparameters for each task. TruthfulQA uses more epochs due to its smaller dataset size.

C.4 Baseline Implementations

Difference-in-Means. For each class c, we compute the centroid of training samples $\mu_c = \frac{1}{|X_c|} \sum_{x \in X_c} h(x)$. Classification assigns the class of the nearest centroid based on cosine similarity.

Linear Probe. A standard linear classifier $W \in \mathbb{R}^{D \times C}$ trained with cross-entropy loss without any constraints on the weight matrix.

Random Token Mixture. To verify whether Token-Mixture's performance stems from meaningful token selection or merely from dimensionality reduction, we implement a baseline that uses K randomly selected tokens instead of the top-K by contribution score. We report averaged results over two random seeds (42, 123) to account for selection variance.

MLP. A two-layer MLP with hidden dimension 256 and ReLU activation, trained with the same optimizer settings as Token-Mixture. This baseline evaluates whether non-linear classifiers provide additional performance gains.

Nonnegative L1. A linear classifier with non-negativity constraints and L1 regularization ($\lambda = 10^{-3}$) operating on the token similarity space $Th(x) \in \mathbb{R}^V$. Unlike Token-Mixture, this baseline does not enforce a hard top-K constraint, allowing us to isolate the effect of explicit token budget limitations.

C.5 Reproducibility

To ensure reproducibility, we fix random seeds for PyTorch, NumPy, and CUDA. All models are ac-

cessed through the following HuggingFace Hub identifiers:

- `meta-llama/Llama-3.1-8B-Instruct`
- `Qwen/Qwen2.5-7B-Instruct`
- `microsoft/phi-4`
- `openai/gpt-oss-20b`

D Additional Results

D.1 User Study for Human Interpretability

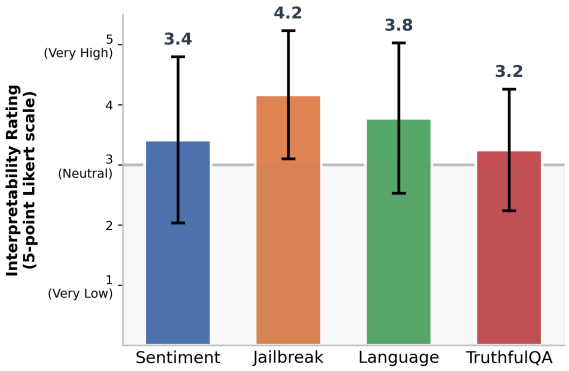

Figure 3: Interpretability ratings on a 5-point Likert scale.

We conducted a user study to evaluate whether Top-K tokens are interpretable to humans. Participants viewed only token lists (without original input text) and performed two tasks: (1) classify the sample, and (2) rate interpretability on a 5-point Likert scale.

As shown in Figure 3, participants achieved **100%** accuracy on Jailbreak and **92%** on Sentiment using tokens alone, with interpretability ratings of 4.2 ± 1.1 and 3.4 ± 1.4 respectively. Language identification was harder (25% accuracy) due to participants' unfamiliarity with non-English tokens, though interpretability remained above neutral (3.8 ± 1.3). Overall helpfulness was rated $3.83/5$, confirming that Top-K tokens provide meaningful, human-interpretable explanations for semantic classification tasks.

D.2 Structural Alignment Analysis with SAE Features

To verify whether Token-Mixture merely exploits lexical regularities of the LM head or genuinely aligns with the model's internal semantic structure, we conduct alignment experiments with monosemantic features from Sparse Autoencoders (SAEs). SAEs are known to extract interpretable single-concept features from LLM internal activations (Cunningham et al., 2024). If Token-Mixture directions exhibit high alignment with SAE features, this would suggest that Token-Mixture captures the model's internal conceptual structure rather than superficial vocabulary patterns.

Experimental Setup. We compute the cosine similarity between Token-Mixture direction vectors $\mathbf{w}_{\text{tm}} \in \mathbb{R}^{4096}$ extracted from Layer 19 of Llama-3.1-8B-Instruct and the decoder matrix $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{65536 \times 4096}$ from the Goodfire SAE.[1] For each semantic axis, we measure the similarity of the most aligned feature among 65,536 SAE features (Max Alignment) and the mean similarity of the top 10 features (Top-10 Mean). For statistical significance testing, we construct a baseline distribution from 1,000 random directions of the same dimensionality and compute z-scores.

Results. Table 7 presents the SAE alignment results for the four semantic classification tasks. Across all tasks, Token-Mixture directions exhibit statistically significant alignment compared to the random baseline (mean z-score = 12.32, $p < 0.001$). Notably, the jailbreak detection task achieves the highest alignment (Max Alignment = 0.168, z-score = 18.74), suggesting that Token-Mixture is strongly connected to the model's internal representations related to safety.

Task	Best Class	Max Align	Top-10 Mean	Z-Score
Jailbreak	jailbreak	0.168	0.142	18.74
TruthfulQA	false	0.141	0.118	13.73
Language	chinese	0.123	0.098	10.38
Sentiment	positive	0.102	0.084	6.44
Average	–	**0.134**	**0.111**	**12.32**

Table 7: Alignment between Token-Mixture semantic directions and SAE monosemantic features. Max Align denotes the maximum cosine similarity among 65,536 SAE features; Z-Score indicates statistical significance relative to the random direction baseline. The mean Max Alignment for the random baseline is 0.068.

Interpretation. These results demonstrate that Token-Mixture is not merely exploiting lexical regularities. Since SAE features are discovered through unsupervised decomposition of model internal activations, the alignment of Token-Mixture directions with these features indicates that two independent methodologies capture similar semantic structures. However, the absolute alignment

[1] https://huggingface.co/Goodfire/Llama-3.1-8B-Instruct-SAE-l19

strength (mean 0.134) is moderate, suggesting that Token-Mixture and SAE features provide complementary perspectives rather than representing identical concepts. This confirms that Token-Mixture maintains its unique advantage of interpretability in vocabulary space while being structurally connected to the internal conceptual structures discovered by SAEs.

E Extended Analysis of Token-Mixture Performance

This appendix provides extended analysis of the Token-Mixture performance results presented in Table 3 of the main text.

E.1 Random Token Selection Ablation

To verify whether Token-Mixture's performance stems from dimensionality reduction effects or from structured token selection, we conduct comparison experiments with Random Token Mixture. Table 8 presents results across multiple random seeds and tasks.

Table 8: Detailed comparison of Random vs. Top-K Token Mixture (K=100). Δ denotes the difference between Top-K and the Random average.

Task	Method	Acc (%)	Ret (%)	Δ
Sentiment	Random (seed 42)	75.60	81.6	
	Random (seed 123)	79.07	85.4	
	Random (avg)	77.33	83.5	
	Top-K	**89.04**	**96.1**	**+11.7**
Jailbreak	Random (seed 42)	89.12	89.1	
	Random (seed 123)	91.34	91.4	
	Random (avg)	90.23	90.3	
	Top-K	**97.68**	**97.7**	**+7.5**
Language	Random (seed 42)	32.56	32.6	
	Random (seed 123)	35.90	36.0	
	Random (avg)	34.23	34.3	
	Top-K	**97.00**	**97.2**	**+62.8**
TruthfulQA	Random (seed 42)	58.24	69.8	
	Random (seed 123)	61.90	74.1	
	Random (avg)	60.07	72.0	
	Top-K	**75.46**	**90.4**	**+15.4**

Interpretation. Across all tasks, Top-K selection achieves significantly higher performance than Random selection (average +24.4%p). The largest gap appears in Language (+62.8%p), where random token selection achieves near-chance performance (20% for 5-class classification), while Top-K accurately identifies language-specific tokens. These results confirm that Token-Mixture is a structured method for selecting semantically important tokens, rather than merely performing dimensionality reduction.

E.2 Cross-Task Token Overlap Analysis

We analyze the degree of overlap between token sets selected for different tasks within the same model using Jaccard similarity. By examining whether token overlap exists across tasks even when using the same tokenizer, we verify whether Token-Mixture selects task-specific tokens.

Table 9: Jaccard similarity between tasks within the same model (K=100). Selected tokens show minimal overlap across tasks even within the same model.

Model	Avg Jaccard	Interpretation
Llama	0.0008	Near zero
Qwen	0.0034	Near zero
GPT-OSS	0.0000	Exactly zero
Phi-4	0.0000	Exactly zero

Key Finding. The near-zero cross-task token overlap despite using the same model and tokenizer demonstrates that Token-Mixture selects **entirely different token subsets** for each task. This implies two important points: (1) the selected tokens are **task-specific**, supporting that they are directly relevant vocabulary for the corresponding semantic classification, and (2) different semantic properties are distributed across distinct regions of vocabulary space, confirming that semantic axes are composed of task-specialized token sets rather than a "universal semantic basis" shared across all tasks.

E.3 Layer-wise Performance Distribution

We analyze how the optimal layer for Token-Mixture varies according to model architecture. Figure 4 shows the layer-wise performance distribution for Llama-3.1-8B-instruct.

Table 10: Summary of optimal layers by task (K=100). Rel. Pos. denotes the relative position as a percentage of model depth.

Task	Llama (32L)	Qwen (28L)	GPT-OSS (24L)	Phi-4 (40L)
Jailbreak	L25 (78%)	L17 (61%)	L14 (58%)	L39 (98%)
Language	L1 (3%)	L3 (11%)	L4 (17%)	L1 (3%)
Sentiment	L10 (31%)	L15 (54%)	L15 (63%)	L38 (95%)
TruthfulQA	L14 (44%)	L9 (32%)	L5 (21%)	L20 (50%)

Observations. Optimal layers vary significantly across tasks. Language classification peaks very

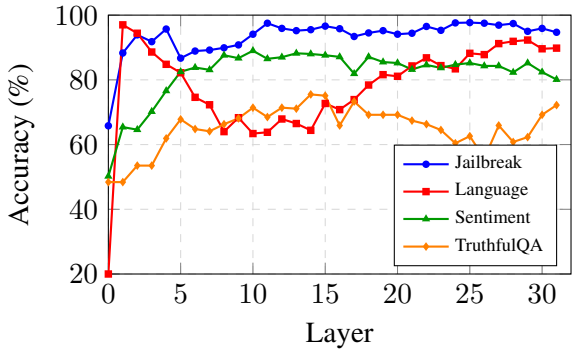

Figure 4: Layer-wise Token-Mixture performance on Llama-3.1-8B-Instruct (K=100). Language peaks early at L1, while Jailbreak peaks late at L25. Sentiment and TruthfulQA show optimal performance in middle layers (L10–L14).

early (L1–L4, 3–17% depth), suggesting language-specific features are encoded in the earliest transformer blocks. In contrast, Jailbreak detection requires deeper layers (58–98% depth), indicating the need for higher-level semantic understanding. Sentiment and TruthfulQA show intermediate patterns with more variation across models.

E.4 Saturation Curve Analysis

We analyze performance saturation patterns as a function of token budget K. Figure 5 shows per-task performance curves and the 95% retention threshold.

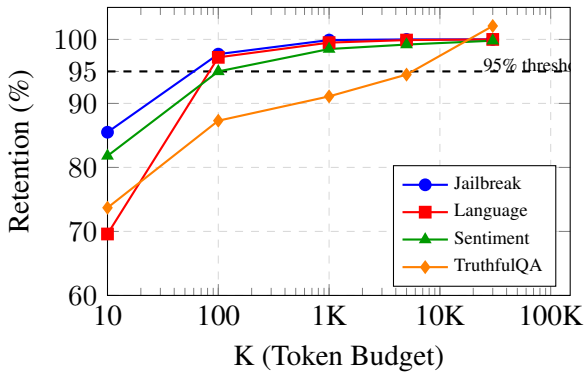

Figure 5: Performance saturation curves by task as a function of K (Llama-3.1-8B-Instruct). Retention denotes the performance ratio relative to full vocabulary. The dashed line indicates the 95% retention threshold. TruthfulQA exceeds 100% at K=30K, indicating that sparse token selection can outperform full vocabulary.

Task Complexity Hierarchy. Saturation K values form a natural hierarchy reflecting the semantic complexity of each task:

Table 11: Minimum K required to achieve 95% retention (Saturation K).

Task	Llama	Qwen	GPT-OSS	Phi-4	Mean
Jailbreak	100	100	100	100	100
Language	100	100	1,000	100	325
Sentiment	100	1,000	1,000	100	550
TruthfulQA	5,000	5,000	30,000	5,000	11,250

1. **Jailbreak (K=100, 0.08% vocab):** Characteristic patterns of adversarial prompts are clearly distinguished by a small number of tokens.

2. **Language (K=325, 0.25% vocab):** Language-specific scripts and function words provide strong classification cues.

3. **Sentiment (K=550, 0.43% vocab):** The diversity of emotional expressions requires broader vocabulary coverage.

4. **TruthfulQA (K=11,250, 8.77% vocab):** Truthfulness detection requires multi-layered signals including entities and discourse patterns, necessitating the highest K.

Conclusion. This analysis demonstrates that Token-Mixture's sparse structure varies systematically according to task characteristics, and that most semantic classification tasks can achieve over 95% performance using less than 1% of the full vocabulary.

Use of AI Assistants

This work utilized AI assistants in the following capacities:

- **Translation:** Translating draft text for manuscript preparation.
- **Code Development:** Assisting with debugging and refining experimental code for probe training and evaluation pipelines.

All research conceptualization, experimental design, data analysis, and scientific conclusions are solely the work of the authors. The authors take full responsibility for the content of this paper.