A Sparse Vocabulary Decomposition of Linear Probe Directions

Anonymous ACL submission

Abstract

Linear probes consistently uncover semantic directions in LLM hidden states—axes separating truth from falsehood, fact from error, safe from harmful. Yet it is rarely asked what those directions *are* in terms the model itself uses. We study this question and find that the probed directions are not spread arbitrarily across the hidden space: they are well approximated by sparse combinations of a small number of the model's own vocabulary tokens. To measure this, we introduce the **Sparse Vocabulary Probe (SVP)**, a linear classifier whose weight is constrained to a K-sparse combination of token-embedding directions, so that each learned direction can be read off directly as a handful of interpretable tokens. Across three models (Llama-3.1-8B-Instruct, Phi-4-14B, Qwen2.5-7B-Instruct) and four tasks (Marks T/F, TruthfulQA, MMLU, JailbreakV), three findings emerge. **(i) The semantic signal is concentrated on a few vocabulary tokens.** One to four atoms reach 80% of free-probe accuracy in 11/12 (model, task) cells; on a random-label control, K=10 SVP stays near chance while the free probe memorizes to 1.00, indicating the match is not explained by probe capacity alone. **(ii) Composite meanings behave additively.** Two single-meaning SVPs trained independently closely match the composite-meaning SVP at the token level (cos = 0.94/0.96/0.98). **(iii) The identified signal aligns with the model's own generation.** On MMLU, items SVP identifies as "known" are recovered by the model's own sampling 10.1–24.4 %p more often than those identified by a free probe. Together, these results indicate that the directions recovered by probing admit a sparse, interpretable, and approximately compositional decomposition over the model's own vocabulary tokens.

1 Introduction

Large language models often fail to put into words what they appear to know. The same fact yields a correct answer in one context and a wrong one in another (Wei et al., 2022); prompting a model to evaluate its own output improves accuracy (Kadavath et al., 2022); and even at the moment a model emits an incorrect token, its hidden state linearly separates truth from falsehood (Burns et al., 2023; Marks and Tegmark, 2024). This gap between what a model outputs and what its activations encode has come to be known as *hidden knowledge* (Orgad et al., 2025; Gekhman et al., 2025): the model knows more internally than it says.

Over the past few years, this phenomenon has been quantified through a consistent linear structure. Unsupervised methods recover activation directions that separate truth from falsehood (Burns et al., 2023); shifting activations along attention-head directions with high truth-classification accuracy improves TruthfulQA performance at inference time (Li et al., 2023); and difference-in-mean probes identify truth directions that causally influence model outputs (Marks and Tegmark, 2024). Linear classifiers on hidden states predict hallucination with high accuracy (Orgad et al., 2025) and recover, on average, 40% more factual knowledge than external token-probability signals—in some cases identifying correct answers that the model fails to produce even once across 1,000 samples (Gekhman et al., 2025).

The common thread is that language models encode task-relevant information in some linear direction of hidden-state space, beyond

what their outputs directly reveal.

But what *is* that direction?

Existing probes treat this direction as a free vector in $\mathbb{R}^d$ and do not describe it in terms the model itself uses. Yet a language model has one set of coordinates always close at hand: its own vocabulary. Inputs are token ids and outputs are distributions over tokens, so the token-embedding matrices are among the few structures the model is guaranteed to use on both its input and output sides. This motivates a concrete hypothesis, which we investigate in this paper: the semantic directions uncovered by linear probing can be approximated by sparse combinations of a small number of the model's own vocabulary tokens, rather than being spread arbitrarily across $\mathbb{R}^d$.

To examine this hypothesis we introduce the *Sparse Vocabulary Probe* (SVP), a linear probe whose weight is constrained to a K-sparse combination of vocabulary-token directions (§3). Because each probe direction is then a weighted sum of a few tokens, it can be read off directly as interpretable vocabulary items. We find that a small K, two orders of magnitude below the hidden dimension d, already matches the accuracy of an unconstrained free probe (§5.1); that this match reflects structure rather than capacity, since at the same accuracy a free probe fits random labels up to 1.00 while SVP stays near chance; that probes for two single meanings, trained independently, closely match the probe for their conjunction at the token level (§5.2); and that the signal SVP identifies is recovered by the model's own sampling more often than the signal a free probe identifies (§5.3).

Our contributions are as follows.

- We show that the semantic directions discovered by linear probing can be closely approximated by K-sparse combinations of the model's own vocabulary tokens, with K far smaller than the hidden dimension.
- We introduce the Sparse Vocabulary Probe (SVP), a K-sparse probe that expresses each learned direction as a small, directly interpretable set of vocabulary tokens.
- We show that composite meanings behave additively on this vocabulary basis: the probe for a conjunction closely matches the sum of the probes for its constituent single meanings.

2 Related Work

Two lines of prior work bear directly on our question—what coordinate system the semantic directions found by probing live on.

Hidden-knowledge probing. A growing body of work shows that LLM hidden states encode more task-relevant information than the model emits in its outputs. Burns et al. (2023) recover linear directions separating truth from falsehood by unsupervised contrastive search; Li et al. (2023) improve TruthfulQA by shifting activations along high-accuracy attention-head directions; Marks and Tegmark (2024) use difference-in-mean probes to identify truth directions that causally affect outputs; and Orgad et al. (2025) show that linear classifiers on hidden states predict hallucination. Gekhman et al. (2025) report that internal probes recover, on average, 40% more factual knowledge than external token-probability signals. These methods train probe weights freely over $\mathbb{R}^d$: the resulting direction is effective for classification, but is not described in the model's own coordinates, and in the regime $d \gg N$ a free probe can also fit arbitrary labelings, so reported accuracy does not by itself separate hidden-state structure from probe capacity (§5.1).

Linear representations and vocabulary space. A separate line of work connects representations to vocabulary space. Park et al. (2024) formalize the linear representation hypothesis in both the output (unembedding) and input (embedding) representation spaces and unify them through a causal inner product; our work shares the high-level aim of describing semantic directions in embedding/unembedding terms, and differs in decomposing *abstract* axes into a *sparse, directly interpretable* set of individual tokens and in examining how such decompositions compose. Logit Lens (nostalgebraist, 2020) and Tuned Lens (Belrose et al., 2023) read vocabulary distributions off intermediate hidden states, using vocabulary as an output-side readout surface

rather than decomposing directions within the hidden state. The mechanistic-interpretability tradition views MLP blocks as operating in vocabulary space (Geva et al., 2021; Meng et al., 2022; Kaplan et al., 2025; Feucht et al., 2024), but has largely focused on concrete entities and lexical items rather than on abstract axes such as truth or safety. We build on these observations and study how abstract semantic axes decompose, as sparse token combinations, on the vocabulary (App. A for an extended discussion).

3 Sparse Vocabulary Probe

To test the hypothesis of §1, we use a linear probe whose weight is constrained to lie in a sparse combination of vocabulary-token directions, rather than ranging freely over $\mathbb{R}^d$. We call it the *Sparse Vocabulary Probe* (SVP). SVP is the instrument for the rest of the paper: with it, §5.1 shows that a handful of token directions already match a standard probe, §5.2 shows that the direction for a composite distinction is closely approximated by the sum of the directions for its parts, and §5.3 shows that the signal SVP isolates surfaces in the model's own generation.

3.1 Method

Let $W_E, W_U \in \mathbb{R}^{|V|\times d}$ be the input and output (unembedding) token-embedding matrices of an LLM with hidden dimension d and vocabulary V, and let $h_\ell(x) \in \mathbb{R}^d$ be the hidden state at the last token position of layer ℓ for an input x. We stack the two matrices into a dictionary $D = [W_E; W_U] \in \mathbb{R}^{2|V|\times d}$; each row of D is the embedding direction of one token, which we call an *atom* (in the sense used in sparse coding). A standard linear probe scores x for class c by $\langle w_c, h_\ell(x)\rangle$ with an unconstrained weight $w_c \in \mathbb{R}^d$; we use such an unconstrained probe—the *free probe* (FP)—as our baseline.

SVP instead constrains each w_c to a combination of at most K atoms. It uses a *shared* atom set $\{t_k\}_{k=1}^K$: the same K atoms serve every class, and class identity is carried only by the per-class coefficients $\alpha_{c,k}$. Sharing the atom set keeps the decomposition comparable across classes—every class is read on the same vocabulary axes—and keeps the parameter count low: SVP has $C \cdot K$ trainable coefficients (20 for a binary task, 40 for a 4-way task), roughly 400× fewer than the free probe's $C \cdot d$.

For class $c \in \{1, \ldots, C\}$ and input x, the parametrization is

$$s_c(x) = \langle w_c,\, h_\ell(x)\rangle = \sum_{k=1}^{K} \alpha_{c,k}\, \langle D_{t_k,:},\, h_\ell(x)\rangle,$$
$$w_c = \sum_{k=1}^{K} \alpha_{c,k}\, D_{t_k,:}, \tag{1}$$

with distinct atoms $t_k \in \{1, \ldots, 2|V|\}$ shared across classes; only the coefficients $\alpha \in \mathbb{R}^{C\times K}$ vary by class. Class probabilities follow from a softmax over the scores, $p(c \mid x) \propto \exp s_c(x)$. Each weight w_c therefore lies in the span of the K selected vocabulary atoms.

Atom selection follows a greedy scheme of the kind used in sparse coding (Olshausen and Field, 1996) and matching pursuit. Algorithm 1 alternates two steps: the current classification residual is back-projected onto the full dictionary D, and the atom whose class-axis gradient has the largest ℓ_2 norm is appended to the shared set; the coefficients are then refit on the selected atoms by L-BFGS against the cross-entropy loss. Intuitively, the back-projected gradient measures how much each candidate token direction could still reduce the loss, so the greedy rule appends the token that the current residual most strongly points to. Full training settings are given in App. B.3.

4 Experimental Setup

We probe three instruction-tuned LLMs — Llama-3.1-8B-Instruct (Dubey et al., 2024), Phi-4-14B (Abdin et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2024) — on four binary or multi-class tasks: Marks T/F factual statements (Marks and Tegmark, 2024), TruthfulQA-MC1 (Lin et al., 2022), MMLU 4-way (Hendrycks et al., 2021), and JailbreakV-28K vs. Alpaca refusal (Luo et al., 2024; Taori et al., 2023). SVP is compared against an unconstrained linear free probe (FP) on a 70/30 split at the validation-best layer per (model, task). Each result has its own control. For **sparsity**, a random-label memorization control (Hewitt-style control task (Hewitt and Liang, 2019; Belinkov, 2022)) separates

Model	Task	K=1	K=2	K=4	K=5	K=8	K=10	K=100	K=1000	Free
	Marks T/F	**83.0**	91.9	93.4	94.8	95.5	96.1	98.8	99.2	99.3
Llama-3.1-8B	TruthfulQA	**70.9**	72.3	75.7	77.7	80.5	78.8	86.0	88.4	87.0
d=4,096	MMLU	29.8	**36.7**	37.6	37.6	37.1	38.0	40.2	42.0	41.6
	Jailbreak	74.4	**83.4**	89.4	93.6	94.8	95.4	99.4	99.6	99.4
	Marks T/F	49.6	55.6	60.0	59.5	60.2	59.5	**91.6**	96.6	96.6
Phi-4-14B	TruthfulQA	**73.3**	75.7	81.8	83.2	85.6	87.0	90.8	90.1	91.1
d=5,120	MMLU	31.8	37.3	**39.8**	41.8	41.1	41.8	46.4	46.9	48.9
	Jailbreak	60.0	75.3	**92.7**	93.5	97.0	97.4	99.3	99.5	99.6
	Marks T/F	51.3	**90.3**	96.6	97.1	96.5	97.5	98.2	98.3	98.0
Qwen2.5-7B	TruthfulQA	64.7	**77.1**	78.4	79.1	81.8	81.8	85.3	86.6	86.0
d=3,584	MMLU	28.7	31.8	**38.9**	39.1	45.1	47.8	43.8	46.4	45.8
	Jailbreak	76.3	**84.8**	94.0	95.5	97.5	98.3	99.8	99.8	99.8

Table 1: **SVP K-sweep vs Free probe.** Cells = best-of-layer SVP test acc(%) on $\ell \in \{8, 12, 16, 20, 24, 28\}$. **Highlighted** = smallest K reaching 80% of Free probe acc; $K \leq 4$ suffices in 11/12 (model, task) cells (Phi Marks T/F outlier, needs K=100). At K=1000, SVP $\geq$ Free in 8/12 cells; the remaining gaps are Phi MMLU (-2.0), Phi TruthfulQA (-1.0), Phi Jailbreak (-0.1), and Llama Marks T/F (-0.1).

signal from probe capacity. For **composition**, we use the Marks T/F cities/larger_than subset as two single meanings (topic $A \perp$ truth B), train six SVPs independently — two single-axis probes v_A, v_B and four cell probes $\vec{\alpha}_{1,\ldots,4}$ on the (topic $\times$ truth) partition — and test three algebraic identities among the resulting probes: sums ($\vec{\alpha}_A + \vec{\alpha}_B$ vs. $\vec{\alpha}_{A \wedge B}$), composite differences sharing one axis (vs. a scaled single), and composite differences sharing no axis (vs. a scaled composite). For **sampling alignment**, we form a four-cell partition (probe $\times$ model $\arg\max$) on MMLU and measure the recovery gap on disagreement subsets (Gekhman et al., 2025). Architecture specs, dataset sizes, training hyperparameters, balanced-subsampling, and the full protocol are in App. B.

5 Results

Using SVP as our instrument, this section reports three findings, on Llama-3.1-8B-Instruct, Phi-4-14B, and Qwen2.5-7B-Instruct. All findings reproduce consistently across the three architectures.

5.1 Sparsity: a few vocabulary tokens match the free probe

We first ask how many vocabulary atoms a probe constrained to vocabulary directions needs in order to match an unconstrained free probe. A free probe learns d weights per class ($d \in \{3{,}584, 4{,}096, 5{,}120\}$); SVP exposes only K coefficients per class, roughly 400$\times$ fewer at K=10. If two probes whose expressive ceilings differ this drastically reach the same accuracy, then the full $\mathbb{R}^d$ freedom is not needed: the semantic signal is concentrated in a low-dimensional subspace spanned by a few vocabulary atoms.

One to four tokens cross the 80% threshold. The highlighted cells in Table 1 mark, per (model, task) pair, the smallest K that reaches 80% of free-probe accuracy (best over $\ell \in \{8, 12, 16, 20, 24, 28\}$). In **11 of 12 cells, $K \leq 4$ vocabulary atoms suffice**; the lone outlier is Phi Marks T/F, which needs K=100. The order of magnitude is the point: against a free probe with 3,584–5,120 weights per class, one to four tokens—about a 10^{-3} fraction of the available dimension—already account for 80% of its accuracy. Were the signal scattered across $\mathbb{R}^d$, accuracy would fall off gradually as K shrinks; instead, most of the gap is closed by K=2, with further atoms acting as coefficient corrections rather than new directions. By K=10 the median gap to the free probe is 3.8 %p across the 12 cells, by K=100 it has effectively closed, and at $K \geq 1000$ SVP matches or exceeds the free probe in 8 of 12 cells (Table 1).

The match is not explained by probe capacity. A random-label control (Table 2) ad-

	Llama-3.1-8B		Phi-4-14B		Qwen2.5-7B		
Task	SVP	Free	SVP	Free	SVP	Free	Chance
Marks T/F	.55	.67	.56	.68	.58	.84	.50
TruthfulQA	.63	**.98**	.64	**1.00**	.64	**1.00**	.50
MMLU	.36	**.92**	.38	**1.00**	.40	**1.00**	.25
Jailbreak	.65	**.97**	.64	.91	.64	**.98**	.50

Table 2: **Probe memorization on a random-label control (K=10).** Train-fit accuracy with random labels (**bold** $\geq$ 0.9). SVP stays near chance (binary .55–.65, MMLU .36–.40) — K=10 atoms cannot fit noise. FP memorizes at .91–1.00 because $d \gg N$. The same real-task accuracy in Table 1 is therefore consistent with capacity for FP, but *not* for SVP.

dresses the reading that the two probes reach the same accuracy simply through raw capacity: the free probe memorizes random labels at 0.91–1.00 on TruthfulQA, MMLU, and Jailbreak (0.67–0.84 on Marks T/F, where L2 regularization at C=0.1 dampens over-fitting on the smallest training set), while SVP at K=10 stays near chance (0.55–0.65 binary, 0.36–0.40 MMLU). SVP at K=10 does not fit the noise; that $K \leq 4$ atoms nonetheless reach the same real-task accuracy regime as a d-dimensional free probe is therefore consistent with a low-rank, vocabulary-aligned structure in the hidden state, rather than with a coincidence of expressive capacity. We note that this control isolates capacity, not the choice of dictionary; whether the vocabulary is special relative to other low-dimensional bases is discussed in the Limitations.

The selected atoms are interpretable tokens. Because each SVP direction is a sum of vocabulary atoms, it can be read off directly. On Llama-3.1-8B-Instruct (ℓ=20, K=10), the SVP truth direction fit on Marks T/F decomposes into `yes`/`Yes`/`YES`/`indeed` on the true pole and `false`/`False`/`untrue` on the false pole. Inputs in Marks T/F are statements such as "The city of Krasnodar is in Russia.", whose next token is whitespace or EOS—not `yes` or `false`. The decomposing tokens are thus not next-token candidates, yet the truth value of the statement is carried on their coordinates: the probe recruits the tokens commonly used to talk about truth.

5.2 Composition: single meanings add up to composites

§5.1 showed that the semantic signal for a single task is concentrated on a few vocabulary atoms. A natural next question is whether *composite* meanings are built from single ones on this same basis. If the model composes single meanings into composite ones, rather than storing composites separately, the trace of that composition should appear on the vocabulary basis as simple arithmetic relations: the sum of two single-meaning probes should match the probe of their conjunction; the difference between a composite and one of its constituents should recover the other constituent; and the difference between two composites should recover a third. None of these relations is enforced by the training objective, so their holding consistently would indicate that the model constructs composite meanings as algebraic combinations of single ones.

Data and setup. The composition experiment uses two subsets of the Marks T/F suite, chosen because they share a true/false labeling but cover unrelated content. cities consists of statements like "The city of Tokyo is in Japan"; larger_than consists of numerical comparisons like "Fifty-one is larger than nine". Each statement carries two labels that vary *independently*. One is its *truth value*: "The city of Tokyo is in Japan" is true, while the otherwise-identical "The city of Tokyo is in China" is false. The other is its *topic*—whether it is a cities or a larger_than statement—and this is unrelated to whether it is true. Pooling the two subsets therefore sorts every statement into one of four groups (Table 4): true and false cities statements, and true and false larger_than statements. We call *topic* and *truth* the two *single meanings*—a topic probe separates cities from larger_than while ignoring truth, and a truth probe separates true from false while ignoring topic—and each of the four groups a *composite meaning* that fixes both at once (e.g. "a true cities statement").

The probes. On Marks T/F we form a balanced four-cell partition over (topic) $\times$ (truth) from the cities and larger_than subsets, and train probes at the (model, layer) of §5.1 in

			Llama-3.1-8B			Phi-4-14B			Qwen2.5-7B		
Expression	**Target**	**Acc. task**	acc	cos	match	acc	cos	match	acc	cos	match
$\vec{\alpha}_{\text{cities}} + \vec{\alpha}_{\text{true}}$	$\vec{\alpha}_1$	cities $\wedge$ true	99.55	0.943	17/19	77.06	0.964	16/19	99.89	0.977	19/20
$\vec{\alpha}_{\text{cities}} + \vec{\alpha}_{\text{false}}$	$\vec{\alpha}_2$	cities $\wedge$ false	99.33	0.933	18/19	76.17	0.982	18/19	100.0	0.978	19/20
$\vec{\alpha}_{\text{lt}} + \vec{\alpha}_{\text{true}}$	$\vec{\alpha}_3$	lt $\wedge$ true	100.0	0.950	18/19	78.17	0.966	**19/19**	100.0	0.970	19/20
$\vec{\alpha}_{\text{lt}} + \vec{\alpha}_{\text{false}}$	$\vec{\alpha}_4$	lt $\wedge$ false	100.0	0.949	**19/19**	84.52	0.982	**19/19**	100.0	0.975	**20/20**

Table 3: **Vocabulary subset arithmetic: sums of singles recover composites.** Pole vectors: $\vec{\alpha}_{\text{cities}}{=}{+}\vec{\alpha}_{\text{topic}}$, $\vec{\alpha}_{\text{lt}}{=}{-}\vec{\alpha}_{\text{topic}}$, $\vec{\alpha}_{\text{true}}{=}{+}\vec{\alpha}_{\text{truth}}$, $\vec{\alpha}_{\text{false}}{=}{-}\vec{\alpha}_{\text{truth}}$; SVPs $\vec{\alpha}_{1,2,3,4}$ defined in Table 4. **acc** (%): 1-D classifier accuracy of the summed vector; **cos**: cosine to target coefficients; **match**: sign-matching atoms (of $|T_A \cup T_B|$). Single-axis $\text{acc}_A/\text{acc}_B$: Llama 100.0/99.7, Phi 97.8/73.8, Qwen 100.0/99.8. Layers ℓ=20 (Llama/Qwen), ℓ=28 (Phi).

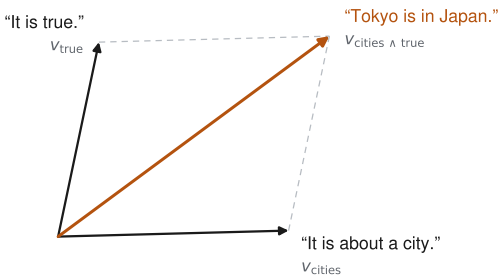

Figure 1: **A statement's meaning is the sum of its single meanings.** The statement *"Tokyo is in Japan"* carries two single meanings—*"it is about a city"* (its topic) and *"it is true"* (its truth value). The probe direction for the composite statement is the vector sum of the probe directions for these two single meanings. Schematic; measured values are in Table 3.

	Truth axis	
Topic axis	$+\vec{\alpha}_{\text{truth}}$	$-\vec{\alpha}_{\text{truth}}$
$+\vec{\alpha}_{\text{topic}}$ (cities)	$\vec{\alpha}_1$	$\vec{\alpha}_2$
$-\vec{\alpha}_{\text{topic}}$ (larger_than)	$\vec{\alpha}_3$	$\vec{\alpha}_4$

Table 4: **Composition probes.** Cell probes $\vec{\alpha}_{1,\ldots,4}$ are fit one-vs-rest on the (topic $\times$ truth) cells of cities$\cup$larger_than; single-axis probes $v_{\text{topic}}, v_{\text{truth}}$ (row/column marginals) define pole vectors $\vec{\alpha}_{\text{cities}}{=}{+}\vec{\alpha}_{\text{topic}}$, $\vec{\alpha}_{\text{lt}}{=}{-}\vec{\alpha}_{\text{topic}}$, etc. All trained independently at the (model, layer) of §5.1.

two stages. **Stage 1.** Single-axis probes $v_{\text{topic}}, v_{\text{truth}}$ are fit by greedy SVP at K=10; their atom indices form the natural basis $T_A \cup T_B$ (size 19/19/20 for Llama/Phi/Qwen). **Stage 2.** The four cell probes $\vec{\alpha}_{1,\ldots,4}$ refit coefficients on this fixed basis by L-BFGS, controlling for vocabulary redundancy (App. B.5). Table 4 defines the layout and pole vectors $\vec{\alpha}_{\text{cities}}, \vec{\alpha}_{\text{lt}}, \vec{\alpha}_{\text{true}}, \vec{\alpha}_{\text{false}}$.

The single meanings occupy nearly disjoint vocabulary supports. The atom sets selected by the two single probes barely overlap: $|T_A \cap T_B| = 1/19$ on Llama and Phi, 0/20 on Qwen. The atoms of v_{truth} index truth value (`false`, `False`, `FALSE`, `/false` on the negative side); the atoms of v_{topic} index the two surface genres (city-name fragments such as `Stanton`, `671`, `Kre` vs. larger_than fragments). No token carries information about both axes: `false` is silent on topic, `Stanton` is silent on truth. Each single meaning lives on a distinct part of the vocabulary.

Sums of singles recover composites. For each of the four cell probes, we compare the sum of the corresponding two single pole vectors ($\vec{\alpha}_{\text{cities}} + \vec{\alpha}_{\text{true}}$, etc.) against the learned composite probe (Table 3, top). The summed coefficients on the natural basis $T_A \cup T_B$ align closely with the learned composite, $\cos(\vec{\alpha}_A + \vec{\alpha}_B, \vec{\alpha}_{A \wedge B})$ = **0.94/0.96/0.98** (Llama/Phi/Qwen), with near-perfect token-by-token sign agreement (Table 3, **match**). Used directly as a 1-D linear classifier, the summed vector reaches essentially the same held-out accuracy as the learned composite probe (Phi lower, in line with its weaker single-task ceiling): the two are functionally equivalent. On Llama the identity also holds at the decision-direction level (Figure 1) — in the plane spanned by v_{topic} and v_{truth}, the sum of

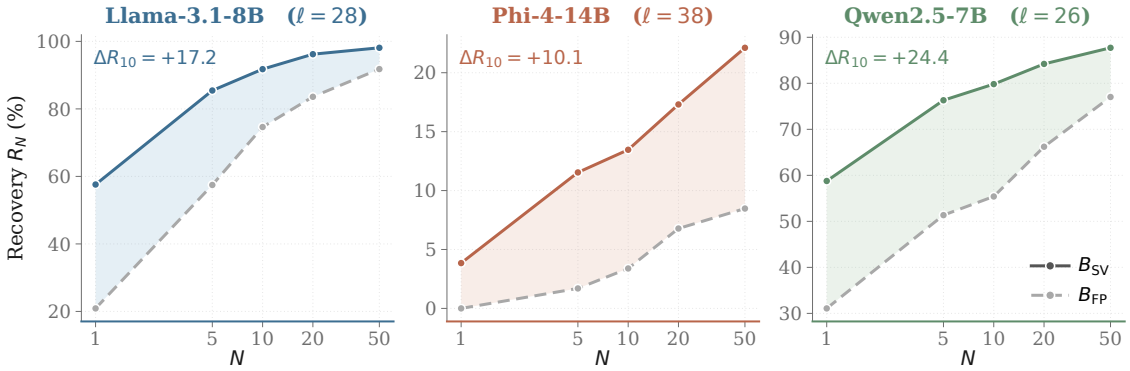

Figure 2: **Disagreement-subset recovery gap.** At each model's SVP-accuracy peak layer, $\Delta R_N = R_N(B_{\mathrm{SV}}) - R_N(B_{\mathrm{FP}}) > 0$ holds for every N (N=10: Llama-3.1-8B-Instruct +17.2 at ℓ=28, Phi-4-14B +10.1 at ℓ=38, Qwen2.5-7B-Instruct +24.4 at ℓ=26, in %p). Hidden knowledge identified by SVP is recovered more often by the model's own sampling than hidden knowledge identified by a free probe.

the two single directions matches the learned composite direction at cos = **1.00**.

Subtraction: singles are recovered from composites. The stronger test is the reverse direction — subtracting two composites should cancel the axis they share and isolate the axis they do not (Table 3, bottom). $\vec{\alpha}_1 - \vec{\alpha}_2$ = (cities $\wedge$ true) − (cities $\wedge$ false) should cancel the cities axis and leave the truth axis: the result aligns with $2\vec{\alpha}_{\mathrm{true}}$ at cos = **0.965/0.913/0.972**. Likewise $\vec{\alpha}_1 - \vec{\alpha}_3$ should cancel the truth axis and leave cities: cos = **0.948/0.955/0.922**. The model does not merely combine singles to form composites; *decomposing* a composite approximately returns the original singles, so composition on this basis is close to invertible.

Composite-to-composite differences and summary. Composite-minus-composite obeys the same arithmetic ($\vec{\alpha}_1 - \vec{\alpha}_4 \approx 2\vec{\alpha}_1$ at cos = **0.999/0.998/0.999**; $\vec{\alpha}_2 - \vec{\alpha}_3 \approx 2\vec{\alpha}_2$ at cos = **1.000/0.995/0.999**; full table in App. C). These identities follow largely from the pole-vector relations above and we report them as an internal consistency check. Taken together, the composite-task probe is well approximated by the sum of its two constituent single-task probes on the vocabulary basis, and the resulting decomposition remains algebraically consistent across composites: the model represents "cities $\wedge$ true not at a separate coordinate but as the sum of the two single meanings it already encodes.

5.3 Sampling alignment: the identified signal surfaces in generation

§5.1 showed that a few vocabulary atoms suffice to classify the semantic signal in the hidden state. Classifiability, however, is measured from the probe's perspective. A complementary question is whether the signal SVP identifies also surfaces in the model's own generation. On MMLU, we compare hidden knowledge identified by SVP against hidden knowledge identified by a free probe, and ask which is recovered more often when the model samples.

Disagreement subset. For each MMLU item, we form the four-case partition (probe prediction) × (model 1-shot arg max), and treat the region where the model is incorrect at 1-shot arg max (model×) as the locus of hidden knowledge. Within this region we isolate the disagreement subset between the two probes:

- B_{SV}: SVP ✓, model × — hidden knowledge identified by SVP;
- B_{FP}: FP ✓, model × — hidden knowledge identified by the free probe.

For each item we sample N continuations at T=0.7 and measure R_N, the fraction of items in which the correct answer appears at least once in the top-N samples. The key metric is the recovery gap

$$\Delta R_N = R_N(B_{\mathrm{SV}}) - R_N(B_{\mathrm{FP}}). \quad (2)$$

A consistently positive ΔR_N indicates that the signal SVP identifies is recovered through the model's own sampling more readily than the signal a free probe identifies.

Headline result. Figure 2 summarizes results across three architectures. The recovery gap is measured at each model's SVP-accuracy peak layer (Llama-3.1-8B-Instruct ℓ=28, Phi-4-14B ℓ=38, Qwen2.5-7B-Instruct ℓ=26). At N=10, the gap is +17.2 %p on Llama-3.1-8B-Instruct, +10.1 %p on Phi-4-14B, and +24.4 %p on Qwen2.5-7B-Instruct. $\Delta R_N > 0$ holds at every N, and even under subset-size imbalance ($|B_{\mathrm{SV}}|/|B_{\mathrm{FP}}|$ = 158/134, 104/59, 114/74), the SVP-identified signal is recovered consistently more often. SVP and the free probe are equivalent from a classifier's perspective, but not from the perspective of the model's own generation.

6 Discussion: interpreting the vocabulary alignment

The three findings of §5 point in the same direction: a few vocabulary atoms suffice to classify the semantic signal, composite meanings are well approximated by sums of single ones on the vocabulary basis, and the signal SVP identifies is recovered by the model's own sampling more often than a free probe's. Here we discuss, more tentatively, why probe directions might align with the vocabulary as cleanly as they do, and what the present results do and do not establish.

One contributing factor is structural. A language model receives its input as token ids and produces its output as a distribution over tokens, so the token-embedding matrices W_E and W_U are the interfaces through which meaning enters and leaves the residual stream. The vocabulary is therefore a coordinate system the model uses at both ends of its computation, which makes it a natural basis on which internal semantic distinctions can be organized.

The interpretability payoff is more concrete: expressing probe directions in vocabulary coordinates yields decompositions that are inspectable as natural-language tokens—e.g. the Llama truth direction's `yes/Yes/indeed` vs. `false/False/untrue` (§5.1)—in a way that free-probe directions in $\mathbb{R}^d$ are not.

7 Conclusion

Linear probing has consistently uncovered semantic directions in LLM hidden states, but those directions have been treated as nameless vectors in $\mathbb{R}^d$, described in coordinates the model itself does not use. We asked whether they instead live on a coordinate system the model is guaranteed to use—its own vocabulary—and introduced the Sparse Vocabulary Probe (SVP) to test this. Across three instruction-tuned models and four tasks, three findings emerge: a handful of vocabulary atoms already match a d-dimensional free probe, with $K \leq 4$ reaching 80% of free-probe accuracy in 11/12 (model, task) cells (§5.1); composite meanings decompose additively into their constituent singles on this same basis, with $\cos(\vec{\alpha}_A + \vec{\alpha}_B, \vec{\alpha}_{A \wedge B})$ = 0.94/0.96/0.98 (§5.2); and the signal SVP isolates surfaces in the model's own sampling 10–24 %p more often than the signal a free probe isolates (§5.3). Taken together, these results indicate that the semantic axes recovered by probing are not arbitrary directions in hidden-state space, but a low-rank, compositional, and interpretable subspace anchored on the model's own vocabulary—a coordinate system on which abstract distinctions such as truth and safety can be both *read* as tokens and *built* from simpler ones.

Limitations

Model scale and architecture. Our experiments are limited to three instruction-tuned dense models: Llama-3.1-8B-Instruct, Phi-4-14B, and Qwen2.5-7B-Instruct. Whether the K-sparse vocabulary decomposition holds with the same strength at frontier scale (100B+), in MoE architectures, or in base (non-instruction-tuned) models is not established here. The scaling tendency itself—whether larger models concentrate meaning on a *sparser* vocabulary subspace, or whether they require *more* atoms—also remains open.

English-only evaluation. All evaluation tasks (Marks T/F, TruthfulQA, MMLU, JailbreakV-28K, Alpaca) are English. Whether the same sparsity and sampling alignment hold on non-English statements,

and how per-language vocabulary subspaces separate or share in multilingual models, are not tested here.

Identifying the role of the vocabulary. Our controls separate signal from probe capacity (§5.1), but they do not isolate the vocabulary dictionary from other low-dimensional bases. We do not compare SVP against a probe constrained to a random dictionary, a PCA basis of the hidden states, or other K-dimensional subspaces. The present results therefore establish that the semantic signal is low-rank and admits an interpretable vocabulary decomposition, but do not by themselves establish that the vocabulary basis is privileged relative to every alternative low-rank basis; we regard a direct dictionary comparison as important future work.

Atom selection and layer dependence. Two further points are not resolved here. First, the greedy atom-selection criterion ranks atoms by gradient norm on an unnormalized dictionary, so it can be sensitive to differences in atom row norm; a length-normalized or whitened selection rule may behave differently. Second, the discussion in §6 does not explain why the vocabulary alignment is already present at the intermediate layers where it is measured, rather than only at the output layer.

Single-seed evaluation. Results are reported for a single train/test split and random seed, without variance estimates or error bars across seeds. In addition, the recovery gap of §5.3 is measured at a layer chosen by SVP accuracy rather than by the validation criterion used elsewhere; a fully independent layer-selection rule, and a control that restricts SVP to input-side (W_E) atoms, would make that comparison cleaner. We regard a multi-seed robustness check as straightforward future work.

Use of AI Assistants

This work utilized AI assistants in the following capacities:

- **Translation and Editing:** Translating draft text and polishing prose for manuscript preparation.
- **Code Development:** Assisting with debugging and refining experimental code for coordinate computation 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.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

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

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*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sheridan Feucht, David Atkinson, Byron C. Wallace, and David Bau. 2024. Token erasure as a footprint of implicit vocabulary items in LLMs. In *EMNLP*.

Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. 2025. Inside-out: Hidden factual knowledge in LLMs. In *COLM*.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *EMNLP*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *EMNLP*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive

multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2025. From tokens to words: On the inner lexicon of LLMs. In *ICLR*.

Shahar Katz, Yonatan Belinkov, Mor Geva, and Lior Wolf. 2024. Backward lens: Projecting language model gradients into the vocabulary space. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Association for Computational Linguistics (ACL)*.

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. *Preprint*, arXiv:2404.03027.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/-false datasets. In *Conference on Language Modeling (COLM)*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

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.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning (ICML)*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Table 5: **Claim–Evidence Map.** Each row maps a core claim to its primary evidence in the main text, the controls, and the robustness checks. "§" refers to main-text sections; letters refer to appendix sections.

ID	Claim	Primary Evidence	Controls	Robustness
C1	**Extreme sparsity.** 1–4 vocab tokens already reach 80% of free-probe accuracy in 11/12 (model, task) cells; a single atom carries non-trivial signal (Llama Marks T/F 83.0%, Phi TruthfulQA 73.3%).	Table 1 (§5.1): K-sweep over $\{1, 2, 4, 5, 8, 10, 100, 1000\}$; highlighted cells = smallest K at 80% FP.	Random-label control (Table 2): K=10 SVP near chance (binary 0.55–0.65, MMLU 0.36–0.40) while FP memorizes (0.91–1.00) $\Rightarrow$ consistent with structure, not capacity.	3 models $\times$ 4 tasks $\times$ 6 layers $\times$ 13 K values (936 cells, App. D); arch.-invariant.
C2	**Sampling alignment.** SVP-identified "known" items surface through sampling more often than FP-identified ones.	Fig. 2 (§5.3): $\Delta R_{N=10} \in [+10.1, +24.4]$ %p.	Probe hits within the model-failure region; subset-size imbalance ($\lvert B_{\mathrm{SV}}\rvert/\lvert B_{\mathrm{FP}}\rvert$ = 158/134, 104/59, 114/74) does not reverse the gap.	3 models at peak layer; $\Delta R_N > 0$ at $N \in \{1, 2, 5, 10, 20, 50\}$.
C3	**Composite = sum of singles.** Two single-task SVPs trained independently combine into the composite-task SVP, atom-by-atom on the vocabulary.	§5.2: $\cos(\vec{\alpha}_A + \vec{\alpha}_B, \vec{\alpha}_{A\wedge B})$ = 0.94/0.96/0.98 (Llama/Phi/Qwen); direction-cos = 1.00 on Llama.	Single-task atom sets nearly disjoint ($\lvert T_A \cap T_B\rvert$ = 1/1/0); each token carries one-axis information only; composite-probe atoms sign-match $T_A \cup T_B$ at 17–20 of 19–20.	3 models consistent; same setup at K=10, ℓ from §5.1.

Appendix Overview

This paper advances three empirically testable claims about the structure of the LLM hidden state on the vocabulary basis. Table 5 provides the full Claim–Evidence Map: each claim is traced to its primary quantitative evidence in the main text, the controls that rule out alternative explanations, and the robustness checks that test generality across models, tasks, layers, and sparsity levels.

Appendix organization. The appendix is structured to support efficient verification of the claims above:

- **App. A** (Extended Related Work) expands the Related Work section (§2) along the two lines of prior work—hidden-knowledge probing and vocabulary-space interpretation—and articulates how this paper moves the discussion from "a linear direction exists" to "that direction is a K-sparse combination of the model's own vocabulary tokens."
- **App. B** (Experimental Protocol & Reproducibility) consolidates all implementation details—model loading, hidden-state extraction, dataset preparation, SVP and free-probe training, balanced sub-sampling for the composition experiment, splits, seeds, and compute—needed for independent replication.
- **App. D** (Full K-Sweep Results) reports the full layer $\times$ K heatmap behind C1, the best-of-layer K-saturation tables for each model.

A Extended Related Work

We expand the Related Work section (§2) along the two lines of prior work that bear directly on our claim.

Hidden-knowledge probing: nameless linear directions. A growing body of work has established that LLM hidden states encode more task-relevant information than the model emits in its outputs. Burns et al. (2023) discovered linear directions that separate truth from falsehood via unsupervised contrastive search (CCS); Li et al. (2023) improved TruthfulQA performance by shifting activations along attention-head directions with high truth-classification accuracy (ITI); Marks and Tegmark (2024) used difference-in-mean probes to identify truth directions that causally influence model outputs; and Orgad et al. (2025) showed that linear classifiers on hidden states strongly predict hallucination. Most recently, Gekhman et al. (2025) demonstrated quantitatively that internal probes recover, on average, 40% more factual knowledge than external token-probability signals, and reported cases where the model *internally* identifies correct answers with near-perfect accuracy that fail to appear even once across 1,000 samples.

The methodological commitment shared across this line of work is to train probe weights freely over all of $\mathbb{R}^d$. The probe makes no use of any structure intrinsic to the model, and the resulting direction is treated as a nameless vector in $\mathbb{R}^d$. This design leaves two questions unanswered. First, there is no instrument with which to ask what that direction *is* in the model's own coordinate system. Second, in the regime $d \gg N$, a free probe is capable of fitting arbitrary labelings, so the reported accuracy does not separate signal coming from hidden-state structure from signal coming from probe capacity (§5.1, random-label control).

Vocabulary-space interpretation: readout surface and concrete entities. Two lines of work have brought vocabulary space to bear on interpretation. Logit Lens (nostalgebraist, 2020) reads vocabulary distributions off intermediate hidden states by applying W_U; Tuned Lens (Belrose et al., 2023) refines this with per-layer affine adapters; and Backward Lens (Katz et al., 2024) extends the same methodology to gradients. This family uses the vocabulary as an *output-side readout surface*; it does not decompose semantic directions inside the hidden state on top of the vocabulary, nor does it train probes directly in vocabulary coordinates.

The mechanistic interpretability tradition, in turn, views MLP blocks as components that operate in vocabulary space. Geva et al. (2021, 2022) interpret FFNs as key-value memories and argue that they promote concepts in the vocabulary space. ROME (Meng et al., 2022) treats the MLP as a linear associative memory and edits factual associations through rank-one updates. Kaplan et al. (2025) show that FFNs reconstruct word-level representations from subword tokens, and Feucht et al. (2024) probe an implicit lexicon through the traces left by token erasure. This line of work establishes that writing into the vocabulary is central to LLM computation—but its targets have been confined to concrete entities and lexical items. How *abstract* semantic axes such as truth, safety, or factual accuracy are represented on top of the vocabulary remains untouched.

Our position. Hidden-knowledge research has established the existence of linear directions but has not asked what coordinate system they sit on; work that engages the vocabulary has used it as a readout surface or as a vehicle for concrete entities, but has not decomposed abstract semantic axes on top of it. This paper moves the discussion from *"a linear direction exists"* to *"that direction admits a sparse, interpretable decomposition over the model's own vocabulary tokens."*

B Experimental Protocol & Reproducibility

This appendix provides the full implementation details needed for independent replication. We describe model loading and representation extraction (§B.1), datasets (§B.2), the Sparse Vocabulary Probe training protocol (§B.3), free-probe and other baselines (§B.4), the balanced-subsample protocol used for the composition experiment (§B.5), and splits / seeds / compute (§B.6). Code and trained probes will be released upon publication.

B.1 Models & representation extraction

Models. All models are loaded from HuggingFace with `torch_dtype="float16"` across 4× RTX 4090

Model	d	Layers	Vocab V
Llama-3.1-8B-Inst.	4,096	32	128,256
Phi-4-14B	5,120	40	100,352
Qwen2.5-7B-Inst.	3,584	28	152,064

Table 6: Architectural specifications of the three LLMs used in all experiments.

Input formatting. The model's official chat template is applied to every input:

```
messages = [{"role": "user",
"content": text}]
formatted =
tokenizer.apply_chat_template(
  messages, tokenize=False,
  add_generation_prompt=True)
```

The last-position hidden state therefore corresponds to the assistant-turn opening.

Hidden-state extraction. For each input x, we forward the model with `output_hidden_states=True` and take $\mathbf{h}_\ell(x) \in \mathbb{R}^d$ at the last token position. Inputs are truncated to 256 tokens (Marks T/F) or 512 tokens (TruthfulQA / MMLU / JailbreakV / Alpaca). The feature-extraction batch size is 8.

Layer selection. For each (model, task) pair, candidate layers $\ell \in \{8, 12, 16, 20, 24, 28\}$ are evaluated, and the layer maximizing validation accuracy at the relevant K is selected. Phi-4 (40 layers) additionally includes $\ell \in \{32, 36, 38\}$ for the recovery analysis of §5.3. All baselines select layer under the same protocol.

Dictionary $D = [W_E; W_U]$. The token-embedding and output-head matrices are cast to `float32` on GPU and row-concatenated to $D \in \mathbb{R}^{2V \times d}$. Rows are not unit-normalized; the L-BFGS coefficient refit handles scale internally. Phi-4 and Qwen2.5 use *untied* embeddings; Llama-3.1 is tied at the weight level (but $D_{t,:}$ for $t \leq V$ and $D_{t+V,:}$ still belong to distinct sides of the residual).

B.2 Tasks & datasets

Table 7 summarizes the dataset configuration. All datasets are class-balanced by subsampling unless otherwise noted.

Task	C	N	Source
Marks T/F	2	1,509	Marks and Tegmark (2024)
TruthfulQA-MC1	2	1,234	Lin et al. (2022)
MMLU 4-way	4	6,000	Hendrycks et al. (2021)
JBV vs. Alpaca	2	6,000	(Luo et al., 2024), (Taori et al., 2023)
Composition (§5.2)			
cities ∪ larger_than (bal.)	4 cells	2,992	Marks and Tegmark (2024)

Table 7: Dataset configuration. **C** = classes; N = items. All splits are 70/30 train/test, class-balanced via sub-sampling. The composition experiment uses a balanced 4-cell (topic × truth) subsample of `cities` and `larger_than` (748 items per cell).

Marks T/F. We combine `cities`, `larger_than`, `sp_en_trans`, and `companies_true_false` from the Marks–Tegmark Geometry-of-Truth suite.

TruthfulQA. We convert TruthfulQA-MC1 into (question, candidate) binary pairs, where each candidate is labeled true / false against its question's reference.

MMLU. We use the standard 4-way multiple-choice format. The model's 1-shot arg max output is used as the prediction baseline for the disagreement-subset analysis of §5.3.

Jailbreak vs. Alpaca. JailbreakV-28K red-team prompts and Alpaca benign prompts are paired in equal numbers, after deduplication on the prompt text.

B.3 Sparse Vocabulary Probe training protocol

Algorithm. Greedy atom selection (by the ℓ_2-norm of the per-class CE gradient back-projected onto D) followed by L-BFGS coefficient refit on the selected atoms (Algorithm 1 in the main text). All training is full-batch.

Table 8 lists all hyperparameters used for SVP and the free probe.

Sparse Vocabulary Probe (SVP)	
Sparsity K (sweep)	$\{1, 2, 4, 5, 8, 10, 20, 50, 100, 200, 500, 1{,}000, 2{,}000\}$
L2 coefficient reg. λ	10^{-2}
Optimizer	L-BFGS / `strong_wolfe`
L-BFGS iter. per step	30
Gradient tolerance	10^{-7}
Atom selection	top-$\|\nabla_D \mathcal{L}_{\text{CE}}\|_2$
Coefficient init.	warm-start prev.
Token exclusion	none
Batching	full batch
Free Probe (FP)	
Classifier	`sklearn.LogReg`
Penalty	L2, C=0.1
Solver	`lbfgs`
`max_iter`	2,000
`fit_intercept`	False

Table 8: Hyperparameters for SVP and the free probe. Both share the same 70/30 train/test split and the same hidden-state extraction protocol (App. B.1).

Algorithm pseudo-code. The greedy selection + L-BFGS refit procedure of §3.1 is summarized below.

Algorithm 1 Sparse Vocabulary Probe

Input: $H \in \mathbb{R}^{n\times d}$, $Y \in \mathbb{R}^{n\times C}$, $D = [W_E; W_U]$, sparsity K, reg. λ

$T \leftarrow \emptyset$; $\alpha \leftarrow \mathbf{0}$
for $k = 1$ **to** K **do**
  $R \leftarrow \text{softmax}(H\, D_T^\top\, \alpha^\top) - Y$ ▷ *residual*
  $G \leftarrow \frac{1}{n} R^\top H\, D^\top$ ▷ *back-project to vocab*
  $t_k \leftarrow \arg\max_{t\notin T} \|G_{:,t}\|_2$ ▷ *greedy atom pick*
  $T \leftarrow T \cup \{t_k\}$
  $\alpha \leftarrow \arg\min_\alpha \text{CE}(\alpha;\, H, Y, D_T) + \lambda\|\alpha\|_F^2$ ▷ *L-BFGS refit*
end for

Output: atoms $T = \{t_k\}_{k=1}^K$, coefficients $\alpha \in \mathbb{R}^{C\times K}$, directions $w_c = \sum_k \alpha_{c,k}\, D_{t_k,:}$

B.4 Free probe and random-label control

Free probe. `sklearn.linear_model.LogisticRegression` with C=0.1, no intercept; see Table 8.

Random-label control (Table 2). The original labels are uniformly permuted with NumPy default RNG `seed=42`. We then fit both SVP (K=10) and the free probe on the (input, random-label) pairs using the same training protocol and report the train-fit accuracy. A trustworthy probe should stay near chance on this control (Hewitt and Liang, 2019; Belinkov, 2022); the gap between SVP (near chance) and the free probe (memorizing 0.91–1.00) confirms that SVP's real-task accuracy is not produced by raw fitting capacity.

B.5 Composition experiment protocol

Balanced 4-cell sub-sample. The Marks T/F subsets `cities` and `larger_than` have 748 and 990 items per 4-cell (topic, truth) in the original release. We sub-sample each cell to 748 items (2,992 total) with NumPy `seed=7`, then apply the same 70/30 split as the main probes.

Natural basis $T_A \cup T_B$. Per-axis SVPs (topic A, truth B) are fit at K=10; their atom indices are unioned to form the natural basis $T_A \cup T_B$. The union sizes are $\{19, 19, 20\}$ for Llama / Phi / Qwen (some atoms are shared between axes).

B.6 Splits, seeds, and compute

Splits and seeds. 70/30 random train/test split (NumPy `seed=0`); `seed=42` for random-atom controls; `seed=7` for the composition balanced sub-sample. Single-seed results are reported throughout, since the K-sweep already provides a stability check over a much wider range than typical seed variation.

Compute. All experiments run on 4× RTX 4090. A single (model, task, K) cell of Table 1 takes 30 s at K=10 and ∼3 min at K=2000; the full K-sweep is ∼1 GPU-hour per model. The composition K-sweep is ∼40 GPU-min per model (one forward cached, then 4 ops × 6 K-values × fixed-union refit).

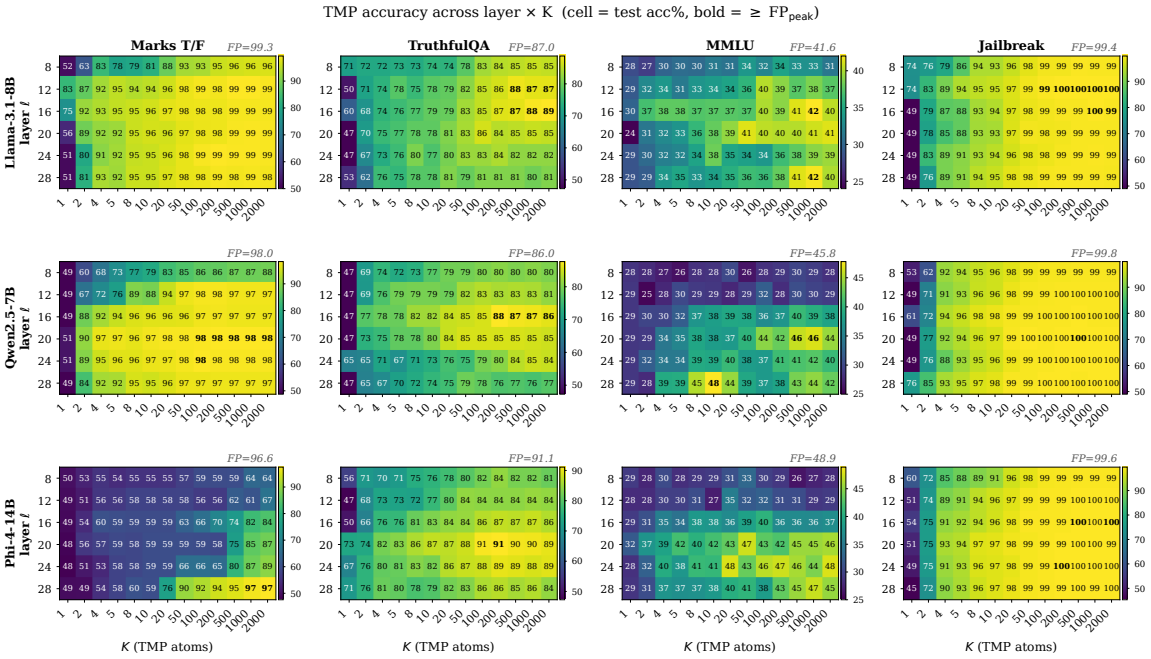

Figure 3: **Full (layer × K) SVP accuracy heatmap.** 3 models (rows) × 4 tasks (columns) = 12 panels; each panel is a 6×13 grid over layers and K. Color = test accuracy (%); **bold** = SVP $\geq$ FP$_{\text{peak}}$ (reported at the top-right of each panel). The saturation region in the layer-K plane differs sharply by task: Jailbreak saturates early (already at K=20, $\ell \geq 16$), while Phi's Marks T/F only reaches FP at $K \geq 1000$.

C Composition: Subtraction Identities

This appendix reports the subtraction half of the composition arithmetic experiment of §5.2; the addition half is in Table 3.

D Full K-Sweep Results

This appendix gives the layer-axis-full view of the K-sweep experiment (§5.1), showing where the best-of-layer summary in Table 1 is achieved and how the saturation profile in K varies layer by layer (Fig. 3). The setup is 3 models × 4 tasks × 6 layers ($\ell \in \{8, 12, 16, 20, 24, 28\}$) × 13 K values ($K \in \{1, 2, 4, 5, 8, 10, 20, 50, 100, 200, 500, 1000, 2000\}$), giving 936 cells in total. Each cell uses the greedy atom selection and L-BFGS refit of §3.1 (30 steps).

Saturation profile across K and model. Tables 10–12 report best-of-layer accuracy for each (task, K) cell on each model (**Bold** = SVP $\geq$ FP$_{\text{peak}}$). Llama and Qwen reach the free probe at $K \sim$10–100; Phi, outside of Marks T/F, saturates even faster. The lone outlier is Phi's Marks T/F, which is stuck around 60% for $K \leq 10$; this plateau breaks at K=20 and reaches 96.6% at K=1000, matching the free probe. The vocabulary decomposition captures the same structure on Phi too—it just needs more atoms to do so for this particular task.

The heatmap in Fig. 3 shows the same pattern in two dimensions: warm cells (high accuracy) form an L-shaped ridge across layers $\ell \in [16, 24]$ that saturates at K ranging from $\sim$20 on Jailbreak to $\sim$1000 on Phi's Marks T/F. Apart from the single Phi/Marks cell, no task-model pair requires a large layer budget once K is sufficient.

			Llama-3.1-8B			Phi-4-14B			Qwen2.5-7B		
Expression	**Target**	**Acc. task**	acc	cos	match	acc	cos	match	acc	cos	match
$\vec{\alpha}_1 - \vec{\alpha}_2$	$2\,\vec{\alpha}_{\text{true}}$	truth	99.55	0.965	14/19	72.05	0.913	16/19	100.0	0.972	19/20
$\vec{\alpha}_1 - \vec{\alpha}_3$	$2\,\vec{\alpha}_{\text{cities}}$	topic	100.0	0.948	18/19	98.33	0.955	17/19	100.0	0.922	17/20
$\vec{\alpha}_1 - \vec{\alpha}_4$	$2(\vec{\alpha}_{\text{cities}} + \vec{\alpha}_{\text{true}})$	cities $\wedge$ true	99.55	**0.999**	**19/19**	77.17	**0.998**	**19/19**	100.0	**0.999**	**20/20**
$\vec{\alpha}_2 - \vec{\alpha}_3$	$2(\vec{\alpha}_{\text{cities}} + \vec{\alpha}_{\text{false}})$	cities $\wedge$ false	99.33	**1.000**	**19/19**	76.50	**0.995**	**19/19**	100.0	**0.999**	**20/20**

Table 9: **Composition: subtraction identities.** Subtracting two composites cancels their shared axis and leaves the other. The last two rows ($\vec{\alpha}_1 - \vec{\alpha}_4$, $\vec{\alpha}_2 - \vec{\alpha}_3$) follow largely by construction from the pole-vector relations and are reported as an internal consistency check; see §5.2.

Task	K=1	K=2	K=4	K=5	K=8	K=10	K=20	K=50	K=100	K=200	K=500	K=1000	K=2000	FP
Marks T/F	83.0	91.9	93.4	94.8	95.5	96.1	97.3	97.8	98.8	99.2	99.2	99.2	98.8	99.3
TruthfulQA	70.9	72.3	75.7	77.7	80.5	78.8	80.8	83.2	86.0	86.6	**87.7**	**88.4**	**88.7**	87.0
MMLU (4w)	29.8	36.7	37.6	37.6	37.1	38.0	39.1	41.1	40.2	40.0	41.1	**42.0**	41.1	41.6
Jailbreak	74.4	83.4	89.4	93.6	94.8	95.4	97.7	98.8	**99.4**	**99.5**	**99.5**	**99.6**	**99.7**	99.4

Table 10: **Llama-3.1-8B** (d=4,096): SVP best-of-layer accuracy(%) across K. **Bold** = SVP $\geq$ FP_{peak}. Layer set $\ell \in \{8, 12, 16, 20, 24, 28\}$.

Task	K=1	K=2	K=4	K=5	K=8	K=10	K=20	K=50	K=100	K=200	K=500	K=1000	K=2000	FP
Marks T/F	49.6	55.6	60.0	59.5	60.2	59.5	76.1	89.7	91.6	93.9	95.1	**96.6**	**97.5**	96.6
TruthfulQA	73.3	75.7	81.8	83.2	85.6	87.0	86.6	87.7	90.8	**91.1**	90.4	90.1	89.4	91.1
MMLU (4w)	31.8	37.3	39.8	41.8	41.1	41.8	47.6	47.1	46.4	47.1	45.8	46.9	47.6	48.9
Jailbreak	60.0	75.3	92.7	93.5	97.0	97.4	98.4	99.2	99.3	**99.6**	**99.6**	99.5	**99.6**	99.6

Table 11: **Phi-4-14B** (d=5,120): SVP best-of-layer accuracy(%) across K. **Bold** = SVP $\geq$ FP_{peak}. Layer set $\ell \in \{8, 12, 16, 20, 24, 28\}$.

Task	K=1	K=2	K=4	K=5	K=8	K=10	K=20	K=50	K=100	K=200	K=500	K=1000	K=2000	FP
Marks T/F	51.3	90.3	96.6	97.1	96.5	97.5	97.8	97.8	**98.2**	**98.3**	**98.3**	**98.3**	**98.2**	98.0
TruthfulQA	64.7	77.1	78.4	79.1	81.8	81.8	84.2	85.3	85.3	**87.7**	**87.0**	**86.6**	**86.0**	86.0
MMLU (4w)	28.7	31.8	38.9	39.1	45.1	**47.8**	44.2	39.8	43.8	42.4	**45.8**	**46.4**	44.2	45.8
Jailbreak	76.3	84.8	94.0	95.5	97.5	98.3	99.3	99.7	99.8	99.8	**99.8**	99.8	99.8	99.8

Table 12: **Qwen2.5-7B** (d=3,584): SVP best-of-layer accuracy(%) across K. **Bold** = SVP $\geq$ FP_{peak}. Layer set $\ell \in \{8, 12, 16, 20, 24, 28\}$.