
Representational Capacity: Geometric Limits on Feature Representation in Transformer Language Models

Alexander Guha
Arizona State University
aguha6@asu.edu

Abstract

Model dimension (d_{model}) is a fundamental hyperparameter in transformer-based language models, yet its role in determining the geometric limits of feature representation remains under-explored. Grounded in the Linear Representation and Superposition Hypotheses—which together propose that models encode features as near-orthogonal directions in the latent space—we develop a quantitative framework for estimating how many such directions a model’s latent space can support. We first establish the embedding matrix as a measurable proxy for the near-orthogonality constraints operating across the latent space, proposing that the boundary between meaningful token relationships and incidental similarity in the pairwise cosine similarity distribution provides a concrete estimate of the model’s accepted deviation ε from perfect orthogonality. Applying this metric across dozens of open-source language models reveals two distinct classes: models with high ε whose embeddings lack near-orthogonal structure, and models with low ε that maintain strict near-orthogonality constraints. We then show that the standard Johnson-Lindenstrauss lemma dramatically underestimates the packing efficiency of trained representations and derive an adjusted capacity formula in which the number of near-orthogonal directions depends on the ratio of vectors to dimensions (k/d) rather than the raw count alone—a single modification that reduces prediction error by two orders of magnitude with no additional free parameters. Combining these results, we define *representational capacity* as a quantitative upper bound on the number of distinguishable directions available for features and embeddings within a model’s latent space. The analysis reveals that capacity is exponentially sensitive to ε , and that larger models tend to favor tighter orthogonality constraints over maximizing raw capacity—a pattern compatible with several explanations (a stability–capacity trade-off, a ceiling on usable concepts, or confounds with overall model scale) that we leave open for future work.

1 Introduction

Model dimension (d_{model}) is one of the hyperparameters that controls a transformer-based language model’s parameter count, and primarily determines the size of the embedding^G and latent space^G within the model¹. In practice, d_{model} is heuristically chosen to scale with the other hyperparameters, generally in powers of 2 to facilitate efficient GPU computation.

Naively, one might expect feature vectors of length d_{model} to use each basis vector for a distinct feature. For example, if $d_{model} = 3$, the vector $[1, 0, 0]$ might represent “Cat”, while $[0, 1, 0]$

Code: <https://github.com/Alex-Guha/representational-capacity>

¹Definitions for terms marked with a superscript ‘G’ can be found in the Glossary (Appendix A).

represents “Dog”—and d_{model} would directly bound the number of features the model could work with. In practice, neural networks have long been understood to use *distributed representations*^G, in which each feature is represented by multiple basis dimensions being active simultaneously, and each basis dimension participates in representing many different features (Bengio et al. [2014]). This leads to *polysemanticity*^G, the phenomenon where individual neurons activate for multiple, seemingly unrelated inputs (Olah et al. [2020]).

A specific instantiation of distributed representations that has gained significant traction is the *Linear Representation Hypothesis*^G (LRH). Following from the introduction of the idea of linguistic regularities in the context of word embeddings (*i.e.* “King – Man + Woman \approx Queen”) (Mikolov et al. [2013]), this idea purports that neural language models broadly tend to represent concepts and features as directions in the latent space. Recent works have provided strong evidence for the existence of such linear feature directions in trained models. Cunningham et al. [2023] employed Sparse Autoencoders (SAEs) to decompose language model activations into sparse, interpretable components, recovering feature directions including “parts of individual names, especially last names” and “legal terms and court case references”. Building on this, Templeton et al. [2024] scaled the approach to Claude 3 Sonnet and extracted millions of monosemantic features—specific entities, code syntax, abstract concepts—including a feature direction for “The Golden Gate Bridge”, and demonstrated that amplifying activations along such directions predictably steers model behavior. Park et al. [2024] complement these empirical findings with a theoretical analysis showing that causal interventions along specific directions can predictably manipulate model behavior, further solidifying the link between linear directions and conceptual representations.

Building on Linear Representations, the *Superposition Hypothesis*^G accounts for polysemanticity by suggesting that neural networks leverage near-orthogonality^G to represent more concepts than the number of available dimensions (Elhage et al. [2022]). This idea is grounded in the Johnson-Lindenstrauss (JL) lemma^G (Johnson and Lindenstrauss [1984]). As detailed in Appendix B, the lemma’s guarantee of distance preservation implies that inner products between unit vectors are also preserved within a small accepted deviation ε , allowing exponentially many near-orthogonal directions to exist in high-dimensional space. According to the Superposition Hypothesis, neural networks exploit this property by representing features as near-orthogonal directions in $\mathbb{R}^{d_{model}}$, allowing the number of representable concepts to grow exponentially relative to d_{model} .

Crucially, the SAE-based studies above not only demonstrate the existence of linear feature representations but also provide direct empirical evidence for superposition in the latent space. Templeton et al. [2024] extracted millions of interpretable features from Claude 3 Sonnet, a model whose latent dimension is orders of magnitude smaller than the number of recovered features. Since these features are represented as directions in a space with far fewer dimensions than features, they must necessarily be arranged near-orthogonally—the geometric hallmark of superposition. This establishes that near-orthogonality in the latent space is not merely a theoretical possibility but an observed property of trained models.

Contributions. This paper investigates the geometric constraints that govern how many features a transformer can represent within its d_{model} -dimensional latent space. If models encode features as near-orthogonal directions as the Superposition Hypothesis proposes and SAE studies empirically support, then the number of such directions is bounded by geometric properties of the space: specifically, its dimension and the tolerance for deviation from perfect orthogonality. Motivated by the relationship between tokenization and embedding structure (discussed in Section 2), we analyze the similarity distribution of the embedding matrix as a method to estimate the accepted deviation ε for near-orthogonality within a model’s latent space. At initialization, the embedding matrix maps orthogonal one-hot vectors into $\mathbb{R}^{d_{model}}$ via random weights, and because random vectors in high-dimensional space are near-orthogonal with high probability—the geometric property underlying the Johnson-Lindenstrauss lemma—the initial embeddings inherit this near-orthogonal structure. Training modifies but largely preserves it: the trained distributions develop an extended right tail corresponding to lexical relationships (morphological variants of the same token) and semantic relationships (conceptually related tokens), while the bulk of unrelated token pairs remains tightly clustered near zero similarity. The boundary between these meaningful relationships and incidental similarity—estimated as $\mu + 2\sigma$ of the distribution—provides a concrete, if heuristic, threshold for ε . Applied across dozens of open-source models, this estimator reveals two distinct classes: high- ε models that lack near-orthogonal embedding structure, and low- ε models that maintain

it. We then show that the standard Johnson-Lindenstrauss bound dramatically underestimates the packing achieved by trained representations, and derive an empirically adjusted formula in which capacity depends on the ratio k/d rather than k alone—a single modification that reduces prediction error by two orders of magnitude with no additional free parameters. Combining these, we define *representational capacity*^G as a quantitative upper bound on the number of distinguishable directions available within the latent space, revealing that the available near-orthogonal directions constitute a shared resource among embeddings, unembeddings, and features, that capacity is exponentially sensitive to ε , and that larger models tend to favor tighter orthogonality over raw capacity.

2 Embeddings

This section establishes embeddings as a measurable proxy for estimating the accepted deviation ε for near-orthogonality^G within a model’s latent space^G. We characterize the post-training similarity distribution of the embedding matrix (including the lexical and semantic relationships that form its tail), propose an estimator for ε , and apply it across dozens of models to reveal two distinct classes.

2.1 Tokenization and the Embedding Space

Tokenization maps each of V vocabulary tokens to a learned d_{model} -dimensional vector via an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d_{model}}$ ^G, equivalent to multiplying a one-hot vector $\mathbf{e}_i \in \mathbb{R}^V$ by \mathbf{E} to obtain $\mathbf{x}_i = \mathbf{E}^\top \mathbf{e}_i$. By construction, the input one-hot vectors are perfectly orthogonal: $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 0$ for all $i \neq j$. Perfect orthogonality, however, requires a space with dimensionality at least equal to the number of vectors— V dimensions for V tokens—and since $d_{model} \ll V$ in practice (typical V is 30,000–130,000 while d_{model} ranges from 768–8,192), \mathbf{E} necessarily projects these one-hot vectors into a much smaller space where strict orthogonality is no longer achievable. At random initialization the resulting embeddings are near-orthogonal with high probability via the Johnson-Lindenstrauss lemma^G, so the embedding matrix can be understood as a compressed representation of vocabulary space; as we will see, training largely preserves this structure.

These embeddings reside in $\mathbb{R}^{d_{model}}$, the same space occupied by all subsequent latent representations: the residual stream means embeddings, intermediate latents^G, and features all coexist in $\mathbb{R}^{d_{model}}$ and are subject to the same geometric constraints. We therefore hypothesize that the near-orthogonality observed in embeddings reflects properties of the broader latent space, with one qualification: as discussed in Appendix C, models with tied embedding and unembedding matrices exhibit different structural properties, suggesting their embeddings may not be representative.

2.2 Post-Training Embedding Structure

For each model, we compute the pairwise cosine similarity $\text{sim}(i, j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle / (\|\mathbf{x}_i\| \|\mathbf{x}_j\|)$ between all token embeddings. Across many trained models, the resulting distributions are tightly centered near zero (Figure 1), closely resembling their random initialization; near-orthogonality is preserved through training.

This preservation is not coincidental: while next-token prediction does not explicitly incentivize near-orthogonality, it also does not require embeddings to collapse onto each other. We hypothesize that the model preserves near-orthogonality as a means of maintaining a structured representational space: if embeddings *lean toward*^G hidden feature directions, then maintaining near-orthogonality helps avoid interference between features (as suggested by the Linear Representation Hypothesis).

Two systematic deviations from random initialization are nonetheless visible on closer inspection (Figure 2): the distributions are shifted slightly to the right of zero (an aversion to negative similarities, plausibly because negative pairwise similarities make the QKV projections work harder to produce useful query-key interactions), and they exhibit slightly asymmetric tails—the right tail extends further than the left, reflecting meaningful relationships between certain token pairs, which we examine next.

2.3 Lexical and Semantic Token Relationships

The extended right tail in the similarity distributions corresponds to token pairs with genuine lexical or semantic relationships. Distinguishing these from incidental similarity is essential: meaningful

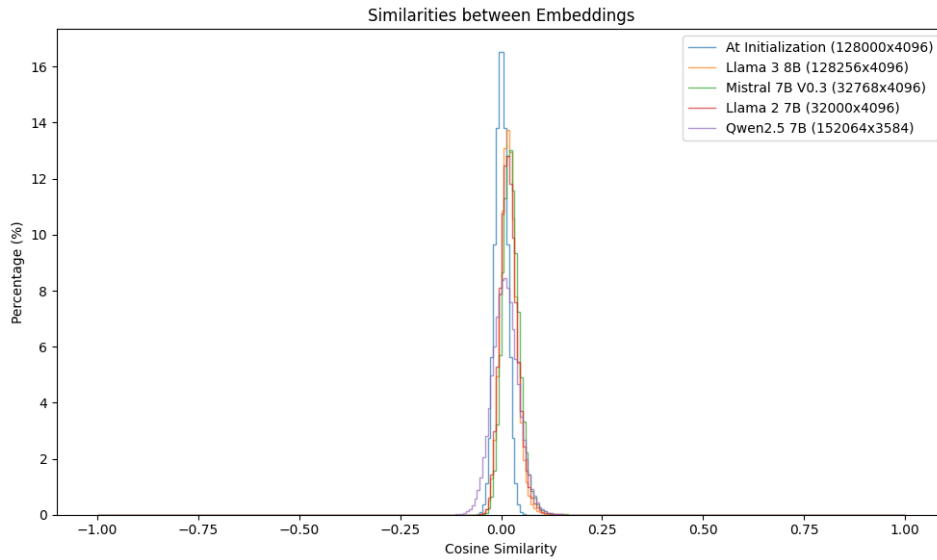


Figure 1: Distribution of pairwise cosine similarity between token embeddings across various models, demonstrating near-orthogonality.

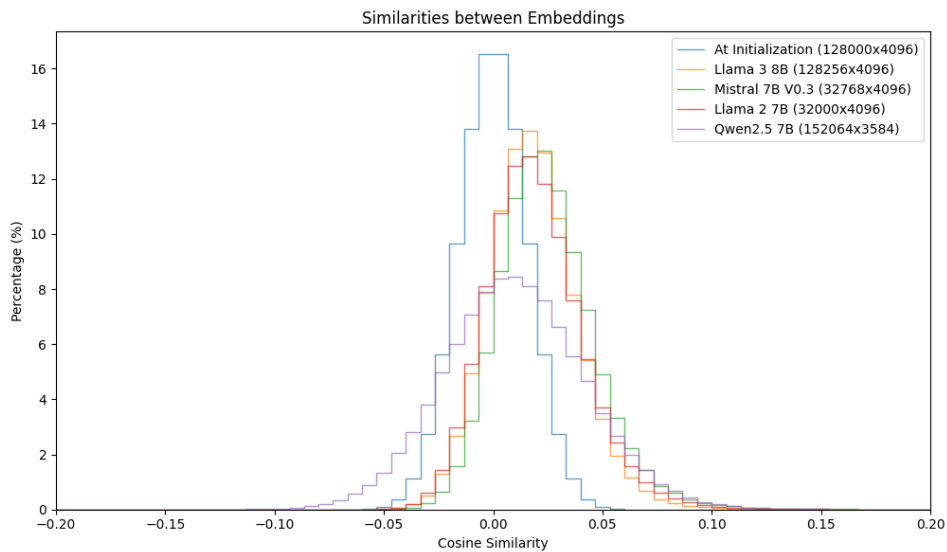


Figure 2: Zoomed view of embedding similarity distributions, revealing the right shift from zero and slightly asymmetric tails.

similarity should not count against orthogonality, while incidental similarity represents the model’s tolerance for feature interference.

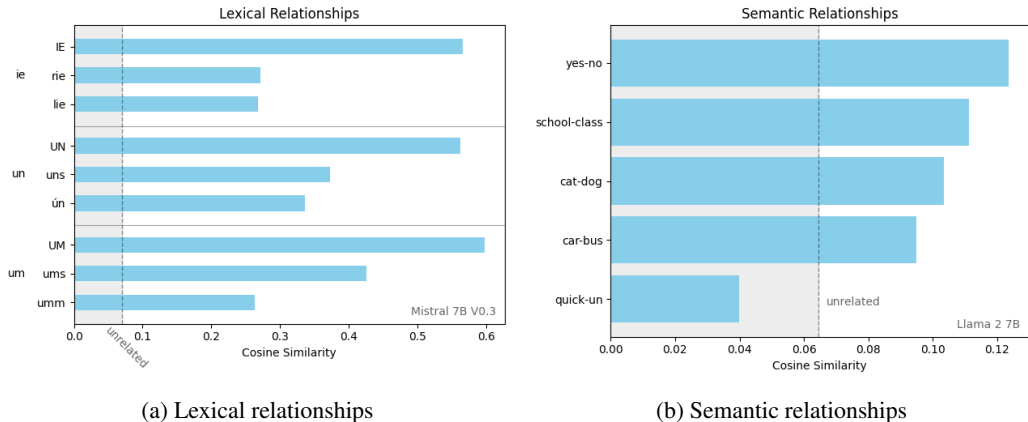


Figure 3: Examples of lexical and semantic relationships in token embeddings. Lexical relationships (a) show tokens with shared surface forms; semantic relationships (b) show conceptually related tokens without lexical overlap. The final pair (quick–un) serves as an unrelated baseline.

Prominent lexical relationships can be identified by examining nearest-neighbor structure. For each token i , we compute its nearest neighbor $\text{nn}(i) = \text{argmax}_{j \neq i} \text{sim}(i, j)$, and call a token k a *primary* token if it appears frequently as the nearest neighbor of others: $|\{i : \text{nn}(i) = k\}| > \tau$ for some threshold τ . These primary tokens (Figure 3a) typically have upwards of 10 secondary tokens closer to them than to any other embedding, and they anchor clusters of capitalization or morphological variants (e.g. “cat”/“Cat”/“cats”) that account for much of the right tail. Semantic relationships (Figure 3b) are more subtle: rather than vectors leaning toward each other directly, conceptually related tokens (e.g. “cat” and “dog”) appear to lean independently toward shared feature directions^G (e.g. “animal”), yielding more limited direct similarity. This distinction remains speculative.

The classic analogy “King – Man + Woman \approx Queen” (Mikolov et al. [2013]) provides further evidence for this semantic structure. Constructing a query $q = x_{\text{king}} - x_{\text{man}} + x_{\text{woman}}$ and retrieving its nearest neighbors among the embedding matrix surfaces both “queen” and “girl” within the top 40 (Figure 4). That “queen” does not rank first likely reflects the imprecision of vector arithmetic as a navigation tool through semantic feature directions; the mere presence of these semantically appropriate tokens corroborates the LRH for decoder-based models—semantic relationships are encoded as directions that can be meaningfully combined.

2.4 Estimating and Generalizing Accepted Deviation ε

Token pairs beyond the boundary of meaningful similarity are near-orthogonal largely as a consequence of high-dimensional geometry, and the boundary itself provides a natural estimate for the model’s tolerance of incidental similarity between unrelated directions. We propose estimating ε as approximately $\mu + 2\sigma$ of the pairwise similarity distribution, where μ and σ are its mean and standard deviation. This threshold is a motivated heuristic rather than a formally justified bound, and is intended to mark the transition from the bulk of unrelated similarities to the relationship-driven tail.

This estimator is grounded in embedding geometry, and extending it to features warrants care: the embedding matrix is a fixed set of learned vectors, while features are directions that may be activated to varying degrees during inference. The connection between the two is nonetheless stronger than mere shared dimensionality. The weight matrices in attention and MLP layers—which perform the only direct interaction with latent vectors outside of normalization—function as collections of feature probes. Each weight vector w_i extracts information from a latent h via the inner product

$$h \cdot w_i = \|h\| \|w_i\| \cos \theta, \tag{1}$$

which is fundamentally a scaled similarity measurement on the same $\mathbb{R}^{d_{\text{model}}}$ space, meaning weight vectors must learn directions that correspond to the features they probe. Individual weight vectors

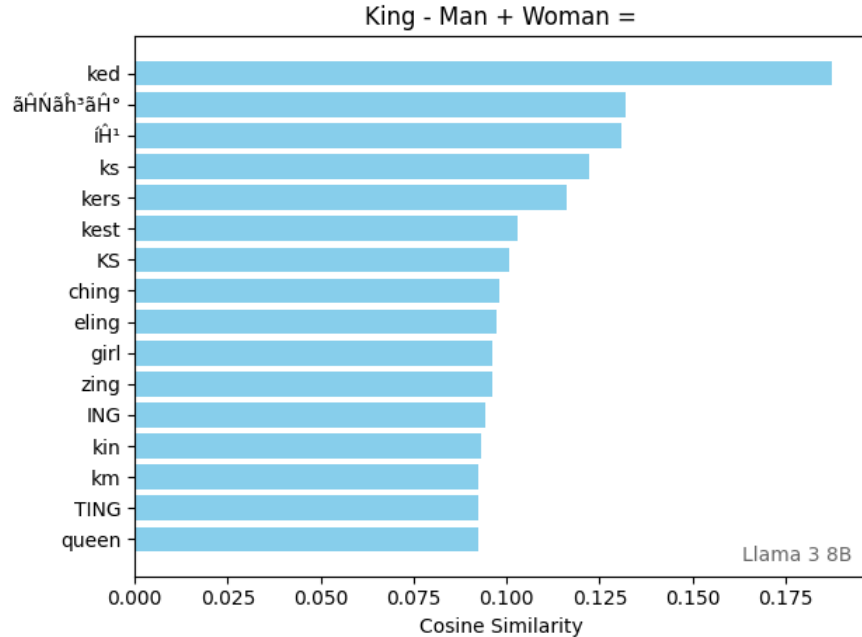


Figure 4: The top 40 most similar embeddings to the vector $x_{\text{king}} - x_{\text{man}} + x_{\text{woman}}$, excluding tokens containing “king”, “woman”, or “women”. The presence of “queen” and “girl” demonstrates semantic structure encoded as directions in the embedding space.

need not align perfectly with any single feature direction—they may represent linear combinations of features, making them difficult to interpret in isolation even when the underlying feature space is well-structured. The outputs of these transformations are then combined with the residual stream by addition, preserving the underlying geometric structure. Nonlinear activations applied after linear transformations can be understood as response functions operating on these similarity scores rather than as modifications to the underlying geometry: directions must first be linearly distinguishable via dot product before nonlinear transformations can selectively act on them. This framing does not preclude features that emerge from multi-layer nonlinear composition, but it does establish that each individual layer’s interaction with the latent space is geometrically constrained by the same near-orthogonality measurement that governs embeddings. Embeddings additionally offer a practical advantage as a fixed, measurable quantity, unlike intermediate representations which vary with input. This generalization nonetheless remains an empirical hypothesis: layer normalization and the specific learned weight configurations could still impose different effective constraints on intermediate representations.

2.5 Two Classes of Models

To validate ε as a meaningful metric, we apply it across dozens of language models and compare against d_{model} . The analysis reveals two distinct classes (Figure 5; per-model values in Appendix D, Table 1), suggesting ε captures a genuine structural property.

The first class consists of models with generally lower d_{model} and wide similarity distributions, yielding $\varepsilon > 0.2$.² These models exhibit similarity-distribution means far from zero (typically 0.1–0.9), the clearest evidence that they do not leverage superposition at the embedding level: when average pairwise similarity is substantially positive, embeddings cannot serve as distinguishable, near-orthogonal directions. Nearly all models in this class have tied embedding and unembedding matrices (see Appendix C).

²An extreme outlier is Gemma 7B, whose ε approaches 1. We do not investigate the cause here, but it appears more consistent with idiosyncratic training or architectural factors than with a deliberate design choice.

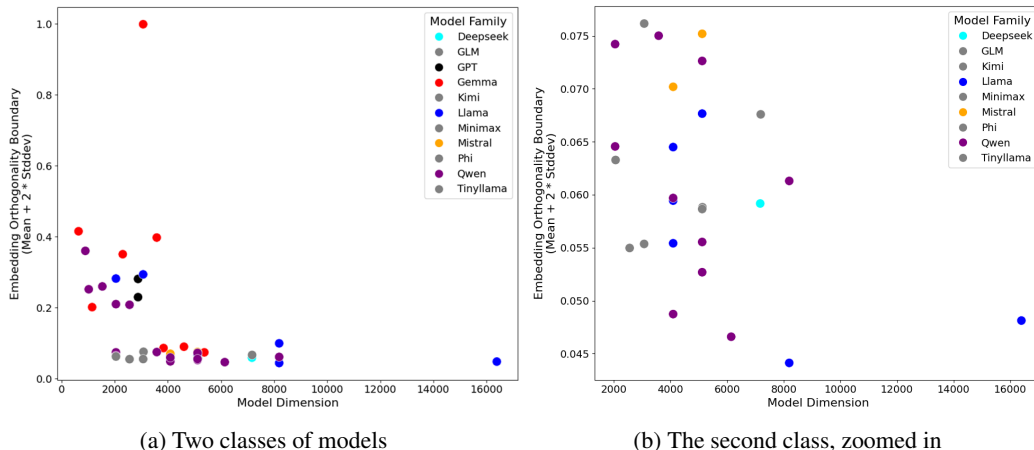


Figure 5: Model dimension compared to estimated ε across various language models, revealing two distinct classes.

The second class consists of models with generally higher d_{model} and tight distributions clustered around zero ($\varepsilon < 0.1$, most around 0.09 or less), consistent with active use of superposition^G to pack many distinguishable directions into the latent space. The cause of the division is not entirely clear: tying correlates with the first class but does not fully explain it, since a few tied models appear in the second class and some untied ones in the first. The remainder of the paper focuses on the second class, where ε -based capacity bounds are meaningful.

2.6 Section Summary

This section established embeddings as a measurable proxy for the near-orthogonality constraints operating in a model’s latent space. At initialization, embedding matrices map orthogonal one-hot vectors into a near-orthogonal configuration in $\mathbb{R}^{d_{model}}$, since random vectors in high-dimensional space are approximately orthogonal with high probability. Training modifies this initial structure only modestly: the distributions shift right (avoiding negative similarities) and develop an extended right tail (encoding lexical and semantic relationships), but the fundamental near-orthogonality persists. The boundary between the main distribution and the relationship-driven tail—estimated as approximately $\mu + 2\sigma$ —provides a concrete threshold for the accepted deviation ε . Under the assumption that embeddings and features coexist in the same d_{model} -dimensional space and are subject to similar geometric constraints, this tolerance plausibly extends to the feature directions that the model learns to use, though this remains a hypothesis rather than a proven relationship.

Applying this metric across dozens of models reveals two distinct classes: high- ε models that do not appear to use superposition at the embedding level, and low- ε models that maintain tight near-orthogonality. Embeddings are therefore not merely a lookup table for token representations. They are a compressed representation of vocabulary space that *leans toward*^G hidden feature directions, placing tokens in superposition^G alongside whatever features the model has learned. Both embeddings and features draw from the same pool of near-orthogonal directions in $\mathbb{R}^{d_{model}}$, subject to the same geometric constraints quantified by ε . The following section uses this ε estimate to derive a quantitative bound on representational capacity.

3 Representational Capacity

We now convert ε into a quantitative bound on *representational capacity*^G—an upper bound on the number of near-orthogonal directions available for features, embeddings, and other learned representations within a model’s latent space. The Johnson-Lindenstrauss (JL) lemma^G provides a natural starting point, but as we show, its random-projection assumption dramatically underestimates the packing achieved by trained models. We derive an empirically adjusted relationship that better matches the geometry of optimized representations and use it to define and compute representational capacity.

3.1 Near-Orthogonal Directions as a Shared Resource

Throughout this section we use k for the number of near-orthogonal directions and $d = d_{model}$ for the latent dimension, consistent with the JL formulation in Appendix B.

Near-orthogonal directions in $\mathbb{R}^{d_{model}}$ serve multiple purposes within a transformer model. Features—the conceptual units the model has learned to represent—occupy directions in this space, as described by the Linear Representation Hypothesis^G. Token embeddings and unembeddings each also require directions: the embedding matrix maps V tokens into $\mathbb{R}^{d_{model}}$, and the unembedding matrix (when untied) maps latent representations back to logits over the vocabulary. For models with separate embedding and unembedding matrices, a rough estimate of the total near-orthogonal directions required is

$$k \approx 2V + k_{features}, \tag{2}$$

where V is the vocabulary size and $k_{features}$ is the number of feature directions. For tied models embeddings and unembeddings share directions, reducing this to approximately $k \approx V + k_{features}$. As noted in Section 2, however, most tied models fall into the high- ε class and may not leverage near-orthogonality at the embedding level; the few tied models in the low- ε class (e.g. Gemma) should be considered with this adjustment.

While we cannot directly measure $k_{features}$, we can work in reverse: given d_{model} and ε , estimate the total k available, and subtract the vocabulary contribution to bound the remaining capacity for features.

3.2 The Johnson-Lindenstrauss Framework and Its Limits

The JL lemma (Appendix B) guarantees that for k unit vectors there exists a linear map to \mathbb{R}^d preserving inner products within $\pm\varepsilon$, provided

$$d \geq C \cdot \frac{\ln k}{\varepsilon^2}, \quad \text{equivalently} \quad k \leq \exp\left(\frac{d \cdot \varepsilon^2}{C}\right), \tag{3}$$

where C depends on the construction. This implies exponential growth of k with d and quadratic growth with ε —the geometric basis for the Superposition Hypothesis^G.

The problem with random vectors. Applied to trained embeddings, however, this bound is wildly off. Using the best proven $C = 8$ for Llama 2 7B ($d_{model} = 4096$, $\varepsilon = 0.0645$) yields

$$k \leq \exp\left(\frac{4096 \times 0.0645^2}{8}\right) = \exp(2.130) \approx 8.4 \tag{4}$$

near-orthogonal directions, against an actual vocabulary of 32,000. To check whether $C = 8$ is simply too conservative, we empirically tighten it. For each (k, d) we generate $T = 1000$ independent trials of k random unit vectors and record the best achievable worst-case similarity:

$$\varepsilon_{\text{random}}^*(k, d) = \min_{t \in \{1, \dots, T\}} \max_{i \neq j} |\text{sim}(\mathbf{v}_i^{(t)}, \mathbf{v}_j^{(t)})|. \tag{5}$$

Fitting (3) to this data yields $C \approx 3.029$ with excellent fit ($R^2 = 0.9985$, MAPE 1.3%, NRMSE 0.9%; Figure 6). Even with this tightened constant, Llama 2 7B is bounded at only $\exp(4096 \times 0.0645^2 / 3.029) \approx 277$ directions—still nearly two orders of magnitude short of its 32,000-token vocabulary. The issue is the assumption rather than the constant: trained embeddings are not random projections, but the result of gradient-based optimization that has discovered arrangements packing far more near-orthogonal directions than random chance allows.

3.3 An Adjusted Relationship for Optimized Vectors

To characterize what optimized vectors achieve, we initialize k random unit vectors in \mathbb{R}^d and minimize a high-exponent penalty $\mathcal{L} = \sum_{i \neq j} |G_{ij}|^p$ on the off-diagonal of the Gram matrix $\mathbf{G} = \mathbf{V}^\top \mathbf{V}$ ($p \in [40, 60]$, selected per scale; Adam optimizer (Kingma and Ba [2017]); 5,000 steps). This implicitly minimizes the maximum pairwise similarity, mimicking the tight packing that gradient descent produces during training. We sweep $d \in$

Standard JL Formula Fit to Random Vector Data

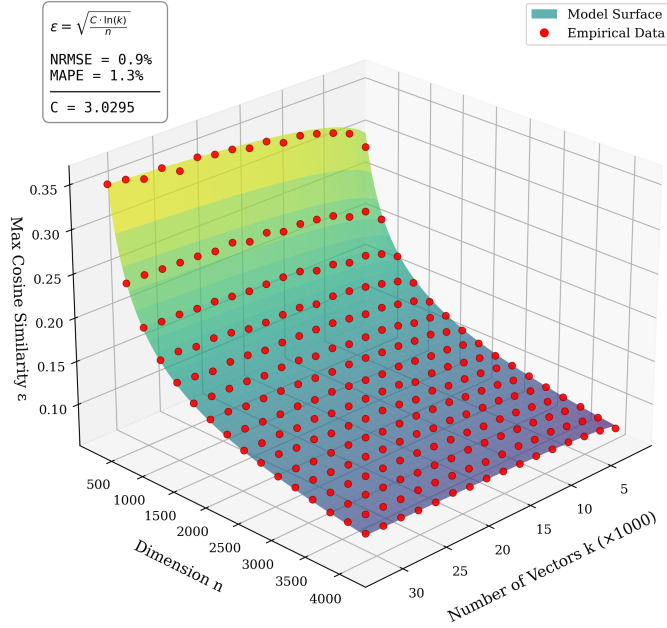


Figure 6: The standard JL relationship $\varepsilon = \sqrt{C \cdot \ln(k)/d}$ fitted to empirically generated random vector data, yielding $C \approx 3.029$ with $R^2 = 0.9985$. Even this tightened constant cannot account for the packing achieved by trained embeddings.

$\{32, 64, 128, 256, 512, 768, 1024, 1536, 2048, 2560, 3072, 3584, 4096\}$ and k from 2,000 to 32,000, incrementing by 2,000 up to 8,000 for all dimensions and by 4,000 from 8,000 to 32,000 for $d \geq 1536$ (the coarser step reflecting the increased cost of optimizing larger configurations), and record $\varepsilon^* = \max_{i \neq j} |G_{ij}|$.

Refitting only the constant C in (3) to this optimized data gives a poor fit (MAPE $\approx 975\%$): the standard JL surface is too flat to track the empirical curve (Figure 7, left). Among many functional-form modifications we tested, replacing $\ln(k)$ with $\ln(k/d)$ produced by far the best alignment:

$$\varepsilon = \sqrt{\frac{C \cdot \ln(k/d)}{d}}, \quad \text{equivalently} \quad k \leq d \cdot \exp\left(\frac{d \cdot \varepsilon^2}{C}\right). \quad (6)$$

Capacity now depends on the *ratio* of vectors to dimensions, not just the raw count. With a single free parameter ($C \approx 1.293$), this fit yields $R^2 = 0.9984$, MAPE 7.9%—a $123\times$ MAPE reduction over standard JL on the same data, with no extra parameters. The factor of d multiplying the exponential is what allows optimized packing to dwarf what random projections can achieve.

A fully parameterized variant $\varepsilon = \sqrt{C \cdot \ln(k^a/d)^b/d^c}$ confirms the structure: fitting yields $a \approx 1.07$, $c \approx 0.97$ (validating k/d and $1/d$ scaling), with only modest gains in fit quality ($R^2 = 0.9998$, MAPE 5.4%; Figure 8). Rearranging for k gives

$$k \leq \left(d \cdot \exp\left(\left(\frac{d^c \cdot \varepsilon^2}{C} \right)^{1/b} \right) \right)^{1/a}. \quad (7)$$

We use this fully parameterized form ($C = 0.458$, $a = 1.067$, $b = 1.447$, $c = 0.972$) for the per-model estimates in Appendix D, but the single-parameter (6) captures the essential structure.

3.4 Defining Representational Capacity

We define the model’s *representational capacity* as the upper bound on k given d_{model} and ε via the adjusted relationship above. For Llama 2 7B ($d_{model} = 4096$, $\varepsilon = 0.0645$) the fully parameterized

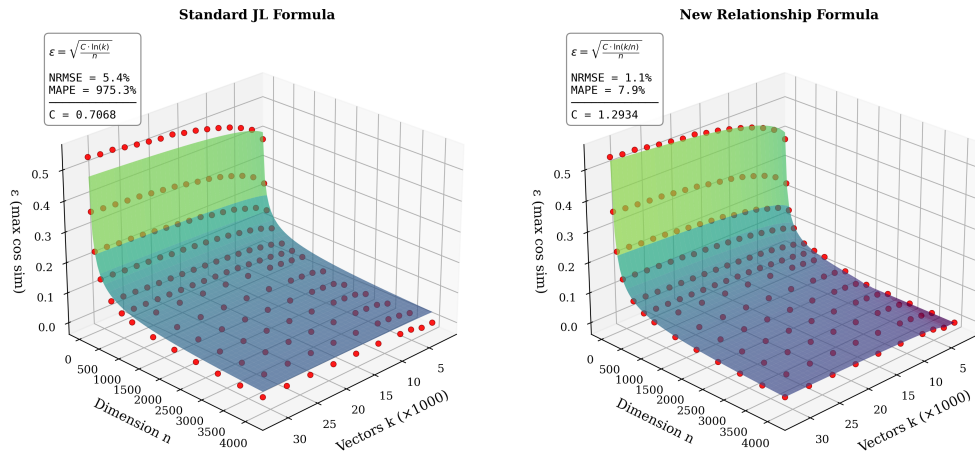


Figure 7: Standard JL formula (left) vs. the adjusted relationship (right) fitted to optimized vector data. Both have one free parameter; the adjusted form fits dramatically better. Red points: empirical data.

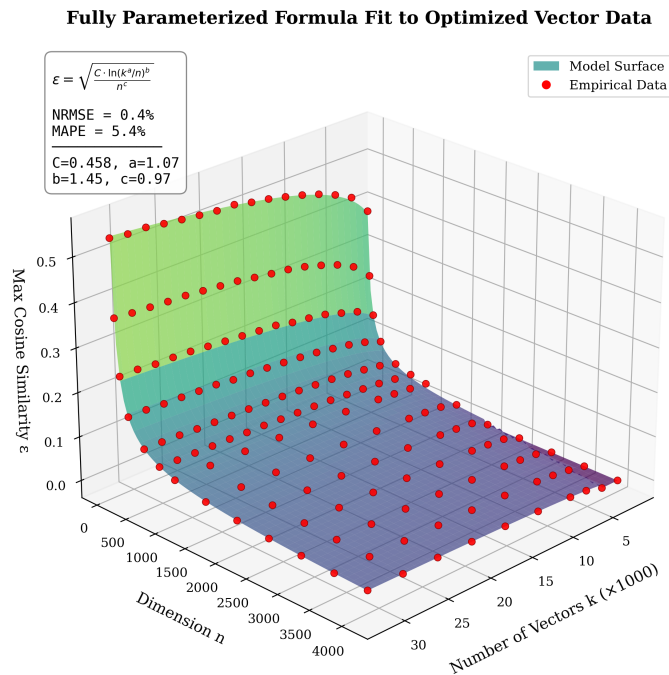


Figure 8: The fully parameterized formula $\epsilon = \sqrt{C \cdot \ln(k^a/d)^b / d^c}$ fitted to optimized vector data, yielding $R^2 = 0.9998$. The modest improvement over the single-parameter form (6) confirms that the k/d ratio is the essential structural insight rather than the extra degrees of freedom.

form gives $k \leq 3.99 \times 10^7$ —roughly 40 million near-orthogonal directions, dramatically higher than the JL estimate of ~ 277 , and comfortably accommodating both the 32,000-token vocabulary (consuming $\sim 64,000$ directions across embeddings and unembeddings) and the millions of features SAE studies have begun to extract.

Two caveats are essential. First, capacity bounds geometric possibility, not realized usage: that 40 million directions *can* exist does not mean the model has learned that many features, nor that its computational architecture (depth, attention heads, MLP width) could effectively utilize them. Second, the ε estimate originates from embedding geometry, and its extension to feature geometry remains a hypothesis (cf. Section 2). Representational capacity is therefore most useful as a *relative* metric: a model with capacity 10^8 has fundamentally more room for features than one with capacity 10^6 , regardless of how many features either actually uses.

3.5 Capacity Across Model Classes

The two-class split from Section 2 extends to capacity. For Class 1 models ($\varepsilon > 0.2$) the framework does not meaningfully apply: their embeddings lack near-orthogonal structure in the first place, so the formula yields vacuous bounds (e.g. $k \leq 2 \times 10^{20}$ at $d_{model} = 2048, \varepsilon = 0.25$). It is possible Class 1 models leverage near-orthogonality at the feature level despite their embeddings showing no such structure, but our embedding-based analysis cannot resolve this. We focus the remaining analysis on Class 2.

For Class 2 models ($\varepsilon < 0.1$), capacities range from $\sim 10^5$ up to $\sim 10^{10}$ – 10^{11} across the models analyzed (Appendix D, Table 1), comfortably accommodating their vocabularies.³ The highest ε we observe within Class 2 is 0.09, with no models approaching the 0.1–0.2 gap from below; this clustering suggests that models may not benefit from, or perhaps cannot sustain, deviations much larger than this threshold while maintaining the structure required for effective superposition. Within this regime, ε exerts a stronger pull on capacity than d_{model} : at fixed $\varepsilon = 0.09$, increasing d_{model} from 2048 to 3072 gives a $30\times$ capacity boost ($2.0 \times 10^7 \rightarrow 6.0 \times 10^8$), but at fixed $d_{model} = 8192$, raising ε from 0.05 to 0.09 gives a six-orders-of-magnitude jump ($2.4 \times 10^8 \rightarrow 2.0 \times 10^{14}$). The exponential dependence on ε^2 makes overlap tolerance, not dimension, the dominant lever.

Despite this leverage, observed models do not fully exploit it. Figure 5b in Section 2 shows a negative correlation between d_{model} and ε within Class 2: larger models maintain tighter near-orthogonality.⁴ This implies a trade-off between raw capacity and internal stability—larger ε admits more features but more interference, while smaller ε sacrifices capacity for more reliable feature retrieval—and suggests that as models scale, they favor stability over capacity maximization. The trend rests on limited data—only a handful of analyzed models have $d_{model} > 6000$, making it difficult to assess whether it continues at larger scales—and the causal mechanism remains unclear. Several explanations are plausible: there may be a practical ceiling on useful capacity beyond which additional feature directions provide diminishing returns; tighter orthogonality may be *necessary* at higher d_{model} to maintain internal stability regardless of capacity benefits; or tighter ε may simply correlate with overall model size, which tends to increase alongside d_{model} in practice. Disentangling these factors would require controlled experiments that vary d_{model} while holding other architectural choices constant, an expensive undertaking that remains beyond the scope of this work.

3.6 Summary

The standard JL lemma, designed for random projections, dramatically underestimates the packing efficiency of learned representations: even with an empirically-tightened constant it bounds Llama 2 7B to fewer than 300 directions, far short of its 32,000-token vocabulary. Replacing $\ln(k)$ with $\ln(k/d)$ in the JL relationship—a single-parameter modification motivated by direct optimization of vector arrangements—reduces prediction error by two orders of magnitude (MAPE 975% \rightarrow 8%) and yields capacity bounds consistent with what is observed in trained models. Applied across models, representational capacity is best understood as a *relative* metric bounding geometric possibility rather than learned utilization: it reveals that Class 1 models fall outside the framework, that Class

³Llama 2 70B is a notable exception, with $\varepsilon = 0.09962$ landing it at $\sim 10^{15}$; this likely reflects idiosyncratic training or architectural factors.

⁴We do not present a separate plot of capacity against d_{model} or ε : since capacity is a deterministic function of the two, such a figure would carry no information beyond Figure 5b.

2 capacities span ten orders of magnitude, and that overlap tolerance ε is the dominant lever, while empirically larger models trade raw capacity for tighter near-orthogonality.

4 Conclusion

Summary. This paper investigated the geometric constraints governing how many features a transformer-based language model can represent within its d_{model} -dimensional latent space, proceeding in four phases across two chapters. In *Section 2*, we first established the embedding matrix as a measurable proxy for the near-orthogonality constraints operating across the latent space, and used the boundary between the bulk of the cosine-similarity distribution and its right tail—estimated as $\mu + 2\sigma$ —as a concrete threshold for the accepted deviation ε . Applying this metric across dozens of models then revealed two distinct classes with no intermediate cases: high- ε models whose embeddings are not approximately orthogonal, and low- ε models that maintain strict near-orthogonality. In *Section 3*, we showed that the standard Johnson–Lindenstrauss framework, designed for random projections, dramatically underestimates the packing efficiency of trained representations—predicting fewer than 300 directions for a model with a 32,000-token vocabulary. By optimizing sets of vectors to mimic what gradient descent implicitly achieves, we derived an adjusted relationship in which capacity depends on the ratio k/d rather than k alone, reducing prediction error by two orders of magnitude with no additional free parameters; we then combined this with the per-model ε estimates to define and compute *representational capacity* as a quantitative upper bound on the distinguishable directions available in the latent space. The resulting picture is one in which embeddings, unembeddings, and features draw from a shared, ε -bounded pool of near-orthogonal directions, and in which larger models tend to favor tighter orthogonality constraints over maximizing raw capacity—suggesting that representational stability may matter more than sheer geometric room at scale.

Limitations. Several caveats apply. (i) The assumption that embedding geometry reflects the broader latent space is a motivated hypothesis, not a proven correspondence: embeddings are a fixed set of learned vectors, while features are dynamically activated directions, and layer normalization plus learned weights could impose different effective constraints over successive layers. (ii) The $\mu + 2\sigma$ threshold for ε is heuristic, and capacity is exponentially sensitive to ε^2 —a shift from $\varepsilon = 0.06$ to 0.09 moves estimated capacity by orders of magnitude—making absolute capacity values unreliable even though relative comparisons remain informative. (iii) Architectural details we did not model (rotary positional encodings on QK subspaces, layer normalization rescaling) may impose distinct constraints on intermediate representations. (iv) Capacity bounds geometric possibility, not realized utilization: the gap between how many directions *can* exist and how many a given architecture can effectively learn and process remains unquantified.

Future work. Several directions follow naturally. The most direct test of our central assumption would measure near-orthogonality in intermediate latent representations across inputs and layers, confirming or refuting whether embedding-derived ε generalizes to the residual stream. Correlating capacity against benchmark performance could reveal whether a “representational scaling law” exists, complementing existing scaling heuristics for choosing d_{model} . A complete scaling theory will likely need to account for both representational capacity (how many features can be stored) and computational capacity (how many can be effectively processed by the network’s depth, width, and attention budget). Finally, the bifurcation into high- and low- ε classes raises open questions: what architectural or training factors cause the divide, is there a critical scale at which models transition into superposition, and do Class 1 models leverage superposition internally despite their embeddings showing no evidence of it?

Implications for model design. Representational capacity currently serves as a diagnostic property of trained models, since ε is not a directly controllable hyperparameter—it emerges from training dynamics and architectural choices such as embedding/unembedding tying. The clustering of Class 2 ε values below 0.09 and the observed tendency of larger- d_{model} models to maintain tighter orthogonality together hint at a possible ceiling on useful capacity, beyond which additional geometric room provides diminishing returns. If such a ceiling exists, d_{model} could in principle be chosen so that the resulting capacity at $\varepsilon \lesssim 0.09$ approximately matches it, avoiding wasted dimensions; conversely, models that have not yet saturated this ceiling may benefit more from increased d_{model} than from other forms of scaling. Either possibility would convert the current heuristic of choosing

d_{model} in powers of two into a principled, capacity-targeted choice—though distinguishing a true ceiling from a stability requirement at scale will require controlled experiments that vary d_{model} while holding other architectural choices fixed.

Closing remarks. We started from a simple question: given a model’s latent dimension, how many features can it represent? The answer depends not just on dimension but on how tightly the model constrains the overlap between its representations—a quantity that emerges from training rather than being set by design. The framework here reframes d_{model} as the determinant of a finite geometric resource that embeddings, unembeddings, and features all draw from, with the model’s tolerance for overlap governing how far that resource stretches. Whether this geometric perspective ultimately connects to model capability—whether capacity predicts what a model can learn, not just what it can store—remains the most compelling open question.

References

- Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. URL <https://arxiv.org/abs/2404.14219>.
- Moonshot AI. Moonshot ai: Kimi intelligent assistant. <https://www.moonshot.cn/>, 2024. Accessed: 2026.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives, 2014. URL <https://arxiv.org/abs/1206.5538>.
- Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL <https://arxiv.org/abs/2309.08600>.
- DeepSeek-AI et al. Deepseek-v3 technical report, 2024. URL <https://arxiv.org/abs/2412.19437>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. URL <https://arxiv.org/abs/2310.06825>.
- William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In Richard Beals, Anatole Beck, and Alexandra Bellow, editors, *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206, Providence, RI, 1984. American Mathematical Society. ISBN 0-8218-5030-X. doi: 10.1090/conm/026/737400. URL <https://archive.org/details/conferenceinmode0000conf/page/189>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL <https://arxiv.org/abs/1412.6980>.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daum   III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL <https://aclanthology.org/N13-1090/>.
- MiniMax. Minimax m2. <https://www.minimaxi.com/news/minimax-m2>, 2025. Proprietary Model Release.
- Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. <https://distill.pub/2020/circuits/zoom-in>.
- OpenAI. Gpt oss: Open source generative pre-trained transformers. <https://huggingface.co/openai>, 2025. Accessed: 2026.
- Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024. URL <https://arxiv.org/abs/2311.03658>.

- Gemma Team and Google DeepMind. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL <https://arxiv.org/abs/2408.00118>.
- GLM Team, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Haoyu Cao, Hanlin Zhao, et al. Glm-4: All tools integrated. *arXiv preprint arXiv:2406.12793*, 2024. URL <https://arxiv.org/abs/2406.12793>.
- Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL <https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html>.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haodong Zhao, Haru Wang, Huanran Wei, Hawen Yin, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL <https://arxiv.org/abs/2407.10671>. Qwen 2.5 is built upon the architecture detailed in the Qwen2 report.
- Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. URL <https://arxiv.org/abs/2401.02385>.

A Glossary

The following glossary provides definitions for key terms and concepts used throughout this paper, particularly those relevant to the analysis of model dimension and representational capacity.

Embeddings The initial vector representations of tokens, produced by the embedding matrix. These are the starting points for the model’s internal processing. Once they pass through the first decoder block, they are no longer considered embeddings and instead become latents.

Latents The vector outputs of the decoder blocks, representing the evolving representations of tokens as they pass through the network. Unlike embeddings, which are static lookups from a learned matrix, latents are dynamically computed based on context.

Note on terminology: Related works often use the term “activations” interchangeably with what this paper refers to as latents. Here, “activations” specifically denotes the output vectors of individual neural layers (e.g., the query vectors resulting from $W_Q X$), whereas “latents” refers to the full residual stream representations after each decoder block.

Latent Space The high-dimensional vector space where the model’s internal representations (embeddings and latents) reside. The dimensionality of this space is determined by the model dimension d_{model} .

Feature/Concept An interpretable unit of information or functionality within the model, such as a specific entity, grammatical rule, or abstract idea. Under the Linear Representation Hypothesis, features are represented as directions in the latent space.

Linear Representation Hypothesis A hypothesis purporting that neural language models represent concepts and features as linear directions in the latent space. Formally, for any concept c , there exists a direction vector $\mathbf{v}_c \in \mathbb{R}^{d_{model}}$ such that the activation or presence of concept c in a latent vector $\mathbf{x} \in \mathbb{R}^{d_{model}}$ is given by the inner product $\langle \mathbf{x}, \mathbf{v}_c \rangle$.

Superposition Hypothesis A hypothesis proposing that neural networks leverage the properties of high-dimensional geometry—specifically near-orthogonality—to represent more concepts than the number of available dimensions. This allows the number of encoded concepts to grow exponentially with d_{model} .

Near-orthogonality A geometric property describing a set of vectors that are approximately orthogonal to each other. Formally, a set of unit vectors $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_k\} \subset \mathbb{R}^d$ is ε -nearly orthogonal if the magnitude of the inner product between any distinct pair is bounded by an accepted deviation ε , meaning $|\langle \mathbf{v}_i, \mathbf{v}_j \rangle| \leq \varepsilon$ for all $i \neq j$, given $0 < \varepsilon < 1$.

Johnson-Lindenstrauss (JL) Lemma A fundamental result in high-dimensional geometry establishing that points in high-dimensional space can be embedded into lower dimensions while approximately preserving pairwise distances and inner products. Critically for the Superposition Hypothesis, the lemma’s distance preservation guarantee implies that orthogonal vectors remain near-orthogonal after projection, enabling high-dimensional spaces to accommodate exponentially many near-orthogonal directions: $k \leq \exp(\frac{\varepsilon^2 d}{C})$ for some constant $C > 0$ (see full derivation in Appendix B).

Polysemanticity The phenomenon observed in neural networks where individual neurons activate for multiple, distinct and often unrelated inputs or concepts (Olah et al. [2020]).

Representational Capacity A quantitative upper bound on the number of near-orthogonal directions available within a model’s latent space, given its model dimension d_{model} and the accepted deviation ε . Embeddings, unembeddings, and features all draw from this shared geometric resource.

Distributed Representations A representation paradigm where each feature is encoded as a pattern of activation across multiple basis dimensions, and each basis dimension participates in representing multiple features (Bengio et al. [2014]).

Note on terminology: In the original distributed representations literature, the term “feature” referred to individual dimensions or neurons in the representation space. Under this paper’s terminology—where “feature” is synonymous with “concept” (see Feature/Concept above)—distributed representations describe how each interpretable concept is spread across many basis dimensions, and conversely, each basis dimension contributes to encoding many concepts.

Leaning Toward A geometric relationship describing when a vector has a higher-than-expected projection onto a particular direction, where the expectation is set by the accepted deviation ε for near-orthogonality. When a vector v leans toward a direction d , the inner product $\langle v, d \rangle > \varepsilon$, indicating meaningful alignment rather than incidental similarity. This concept is central to understanding how embeddings encode information about features while remaining near-orthogonal to unrelated directions.

B Johnson-Lindenstrauss Lemma

Theorem B.1 (Johnson-Lindenstrauss Lemma, Angle Form). *Let $0 < \delta < 1$ and let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ be a set of unit vectors in a (possibly infinite-dimensional) Hilbert space. There exists a linear map*

$$f : \mathbb{R}^N \rightarrow \mathbb{R}^d \quad \text{with} \quad d = O(\delta^{-2} \log k),$$

such that for all i, j ,

$$|\langle f(\mathbf{x}_i), f(\mathbf{x}_j) \rangle - \langle \mathbf{x}_i, \mathbf{x}_j \rangle| \leq 3\delta.$$

In particular, if $\mathbf{x}_i \perp \mathbf{x}_j$, then

$$|\langle f(\mathbf{x}_i), f(\mathbf{x}_j) \rangle| \leq 2\delta,$$

so the images are near-orthogonal.

Proof. We begin with the standard Johnson-Lindenstrauss lemma. There exists a linear map f such that for all $\mathbf{u}, \mathbf{v} \in X \cup \{\mathbf{0}\}$,

$$(1 - \delta)\|\mathbf{u} - \mathbf{v}\|^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq (1 + \delta)\|\mathbf{u} - \mathbf{v}\|^2. \quad (8)$$

Step 1: Norm preservation. Setting $\mathbf{v} = \mathbf{0}$ in (8) yields

$$(1 - \delta)\|\mathbf{u}\|^2 \leq \|f(\mathbf{u})\|^2 \leq (1 + \delta)\|\mathbf{u}\|^2.$$

Since each \mathbf{x}_i is a unit vector,

$$1 - \delta \leq \|f(\mathbf{x}_i)\|^2 \leq 1 + \delta. \quad (9)$$

Step 2: Distance between two unit vectors. For any unit vectors \mathbf{u}, \mathbf{v} ,

$$\|\mathbf{u} - \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\langle \mathbf{u}, \mathbf{v} \rangle = 2 - 2\langle \mathbf{u}, \mathbf{v} \rangle.$$

Applying (8),

$$2(1 - \delta)(1 - \langle \mathbf{u}, \mathbf{v} \rangle) \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq 2(1 + \delta)(1 - \langle \mathbf{u}, \mathbf{v} \rangle). \quad (10)$$

Step 3: Recover inner products using polarization. For any vectors \mathbf{a}, \mathbf{b} ,

$$\langle \mathbf{a}, \mathbf{b} \rangle = \frac{\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a} - \mathbf{b}\|^2}{2}.$$

Applying this to $f(\mathbf{u}), f(\mathbf{v})$,

$$\langle f(\mathbf{u}), f(\mathbf{v}) \rangle = \frac{\|f(\mathbf{u})\|^2 + \|f(\mathbf{v})\|^2 - \|f(\mathbf{u}) - f(\mathbf{v})\|^2}{2}. \quad (11)$$

Step 4: Bounding the inner product distortion. Using (9) and the upper bound from (10),

$$\begin{aligned} \langle f(\mathbf{u}), f(\mathbf{v}) \rangle &\geq \frac{2(1 - \delta) - 2(1 + \delta)(1 - \langle \mathbf{u}, \mathbf{v} \rangle)}{2} \\ &= (1 - \delta) - (1 + \delta) + (1 + \delta)\langle \mathbf{u}, \mathbf{v} \rangle \\ &= (1 + \delta)\langle \mathbf{u}, \mathbf{v} \rangle - 2\delta. \end{aligned}$$

Similarly, using the lower bound from (10),

$$\begin{aligned} \langle f(\mathbf{u}), f(\mathbf{v}) \rangle &\leq \frac{2(1 + \delta) - 2(1 - \delta)(1 - \langle \mathbf{u}, \mathbf{v} \rangle)}{2} \\ &= (1 + \delta) - (1 - \delta) + (1 - \delta)\langle \mathbf{u}, \mathbf{v} \rangle \\ &= (1 - \delta)\langle \mathbf{u}, \mathbf{v} \rangle + 2\delta. \end{aligned}$$

Thus,

$$|\langle f(\mathbf{u}), f(\mathbf{v}) \rangle - \langle \mathbf{u}, \mathbf{v} \rangle| \leq 2\delta + \delta|\langle \mathbf{u}, \mathbf{v} \rangle| \leq 3\delta,$$

where the last inequality uses $|\langle \mathbf{u}, \mathbf{v} \rangle| \leq 1$ for unit vectors.

Step 5: Near-orthogonality. If $\mathbf{u} \perp \mathbf{v}$, then $\langle \mathbf{u}, \mathbf{v} \rangle = 0$, and the bounds in Step 4 give

$$|\langle f(\mathbf{u}), f(\mathbf{v}) \rangle| \leq 2\delta.$$

Hence the images are near-orthogonal.

Step 6: Angles. Since $\|f(\mathbf{u})\|, \|f(\mathbf{v})\| \in [\sqrt{1-\delta}, \sqrt{1+\delta}]$, the cosine of the angle θ' between $f(\mathbf{u})$ and $f(\mathbf{v})$ satisfies

$$|\cos \theta'| = \frac{|\langle f(\mathbf{u}), f(\mathbf{v}) \rangle|}{\|f(\mathbf{u})\| \|f(\mathbf{v})\|} \leq \frac{3\delta}{1-\delta} = O(\delta).$$

This completes the proof. □

Remark B.1. The dimension requirement $d = O(\delta^{-2} \log k)$ can be written as $d \geq C' \cdot \frac{\ln k}{\delta^2}$ for some constant $C' > 0$. For orthogonal vectors, Step 5 shows the inner product deviation is bounded by 2δ . Setting $\varepsilon = 2\delta$ —where ε is the accepted deviation on near-orthogonality used throughout this paper—gives

$$d \geq C \cdot \frac{\ln k}{\varepsilon^2}$$

for an adjusted constant C that absorbs the factor of 4. Rearranging yields

$$k \leq \exp\left(\frac{\varepsilon^2 d}{C}\right),$$

which bounds the number of ε -near-orthogonal unit vectors that a random projection can produce in \mathbb{R}^d . This exponential growth in k with respect to d is central to the Superposition Hypothesis.

The value of C depends on the specific probabilistic construction used; Section 3 discusses both the theoretical value $C = 8$ from the literature and an empirically fitted value for optimized vectors.

C Embedding and Unembedding Relationship

This appendix examines the relationship between embedding and unembedding matrices, which informs the interpretation of embedding-based ε estimates in Section 2 and motivates the qualification noted there for tied models.

At the output layer, the final latent representations are converted to logits via the unembedding matrix $\mathbf{U} \in \mathbb{R}^{V \times d_{model}}$, with the logit for token i given by the inner product $\langle \mathbf{h}, \mathbf{u}_i \rangle$ between the final latent \mathbf{h} and the corresponding row \mathbf{u}_i . In *untied* models, \mathbf{E} and \mathbf{U} are learned independently and may differ significantly; in *tied* models the same matrix serves both roles.

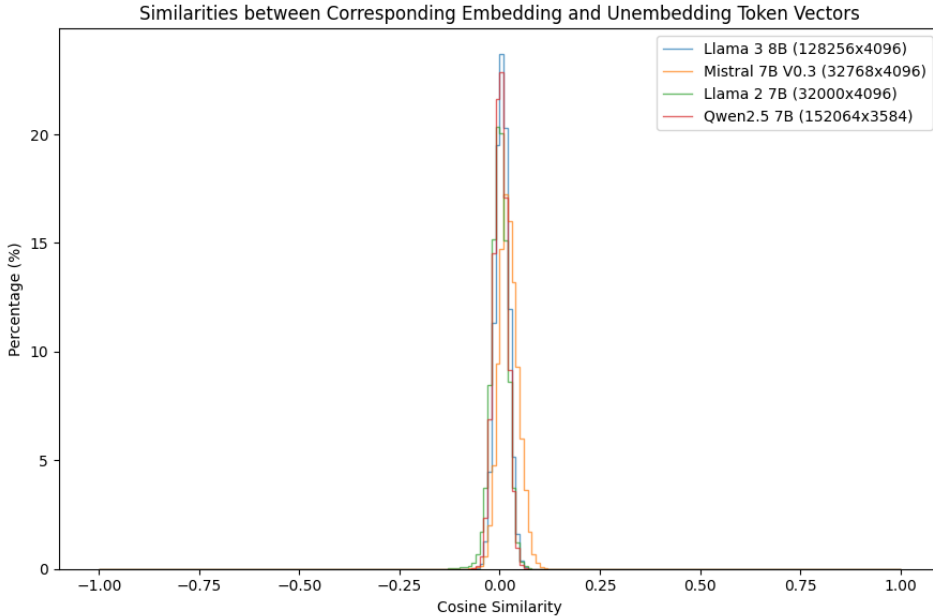


Figure 9: Cosine similarity between corresponding token embeddings and unembeddings for several models with untied matrices.

Comparing the embedding and unembedding vectors for each token in untied models (Figure 9) reveals that they are largely orthogonal: the model transforms representations from an input embedding space into a distinct output unembedding space over the course of the residual stream. The similarity distributions of embeddings (Section 2) and unembeddings (Figure 10) also differ noticeably, likely reflecting the different roles these matrices play—embeddings provide a useful starting representation for contextualization, while unembeddings must support accurate next-token prediction.

In tied models the embedding similarity distributions follow the unembedding pattern, since the two are the same matrix. This is relevant to the two-class split in Section 2: nearly all Class 1 (high- ε) models are tied, suggesting that forcing one matrix to satisfy both objectives discourages the near-orthogonal structure that untied embeddings exhibit. Tying does not fully account for the divide—a few tied models maintain low ε and some untied ones do not—so the structural cause remains open.

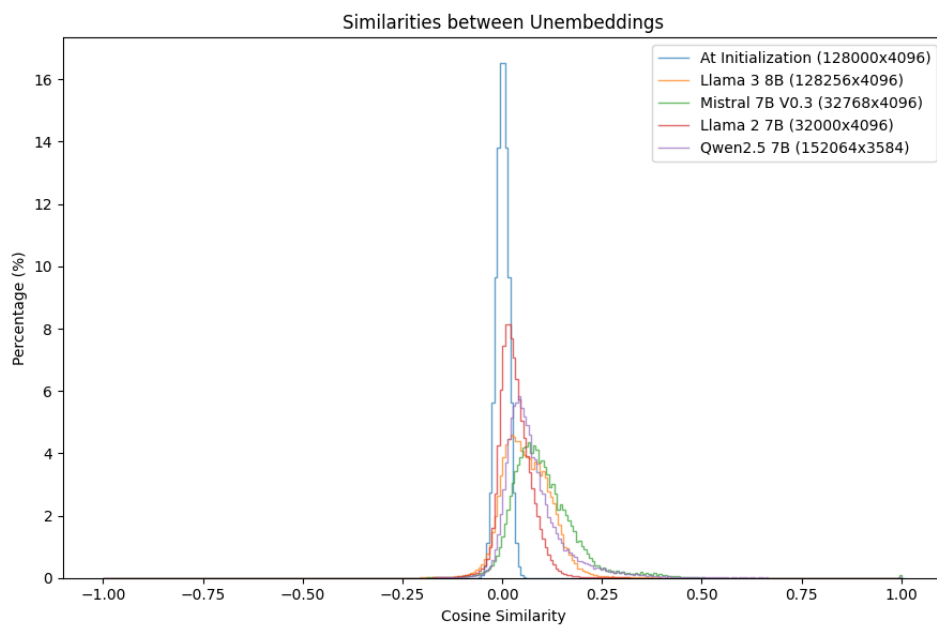


Figure 10: Distribution of cosine similarity between each token unembedding and all others.

D Models Analyzed

Per-model values of d_{model} , the estimated accepted deviation ε ($\mu + 2\sigma$ of the pairwise embedding cosine similarity distribution), and the resulting representational capacity computed via the fully parameterized form of (6) (Section 3.3). Class 1 models ($\varepsilon > 0.2$) are marked indeterminate: their embeddings lack near-orthogonal structure, so the bound is vacuous (cf. Section 3.5). Across the Class 2 models, the bulk of capacities top out near $\sim 10^{10}$ – 10^{11} ; the lone exception is Llama 2 70B, whose $\varepsilon = 0.09962$ sits at the very edge of the Class 1 boundary and lands its capacity at $\sim 10^{15}$.

Table 1: Per-model d_{model} , accepted deviation ε , and representational capacity.

Model Name	d_{model}	ε	Rep. Capacity
DeepSeek V3.2 Exp	7168	0.05917	1.16e9
DeepSeek R1	7168	0.05917	1.16e9
Gemma 7B	3072	0.99918	indeterminate
Gemma 2 2B	2304	0.35084	indeterminate
Gemma 2 9B	3584	0.39785	indeterminate
Gemma 2 27B	4608	0.09013	4.77e10
Gemma 3 270M	640	0.41554	indeterminate
Gemma 3 1B	1152	0.20164	indeterminate
Gemma 3 12B	3840	0.08635	2.51e9
Gemma 3 27B	5376	0.07421	4.36e9
GLM 4.6	5120	0.05882	6.14e7
GPT OSS 120B	2880	0.22984	indeterminate
GPT OSS 20B	2880	0.28120	indeterminate
Kimi K2	7168	0.06763	1.47e10
Llama 2 7B	4096	0.06450	3.99e7
Llama 2 13B	5120	0.06766	5.11e8
Llama 2 70B	8192	0.09962	8.15e15
Llama 3 8B	4096	0.05945	1.42e7
Llama 3.1 8B	4096	0.05542	6.37e6
Llama 3.1 70B	8192	0.04413	4.39e7
Llama 3.1 405B	16384	0.04813	1.22e11
Llama 3.2 1B	2048	0.28223	indeterminate
Llama 3.2 3B	3072	0.29395	indeterminate
MiniMax M2	3072	0.07618	4.39e7
Mistral Small 3.2 24B	5120	0.07520	3.39e9
Mistral 7B Instruct v0.3	4096	0.07019	1.33e8
Phi 2	2560	0.05497	4.57e5
Phi 3 mini 128k	3072	0.05536	1.21e6
Phi 4	5120	0.05865	5.91e7
Qwen 2.5 0.5B	896	0.36035	indeterminate
Qwen 2.5 1.5B	1536	0.25998	indeterminate
Qwen 2.5 3B	2048	0.21014	indeterminate
Qwen 2.5 7B	3584	0.07501	1.20e8
Qwen 2.5 14B	5120	0.05269	1.51e7
Qwen 2.5 32B	5120	0.05554	2.88e7
Qwen 2.5 72B	8192	0.06131	8.46e9
Qwen 3 0.6B	1024	0.25204	indeterminate
Qwen 3 4B	2560	0.20834	indeterminate
Qwen 3 8B	4096	0.05969	1.49e7
Qwen 3 32B	5120	0.07263	1.77e9
Qwen 3 30B A3B	2048	0.06456	5.67e5
Qwen 3 Next 80B A3B	2048	0.07422	2.07e6
Qwen 3 235B A22B	4096	0.04874	1.77e6
Qwen 3 Coder 480B	6144	0.04660	1.21e7
TinyLlama 1.1B	2048	0.06329	4.81e5