CORRELATIONS IN THE DATA LEAD TO SEMANTICALLY RICH FEATURE GEOMETRY UNDER SUPERPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in mechanistic interpretability have shown that many features represented by deep learning models can be captured by dictionary learning approaches such as sparse autoencoders. However, our understanding of the structures formed by these internal representations is still limited. Initial “toy-model” analyses showed that in an idealized setting features can be arranged in local structures, such as small regular polytopes, through a phenomenon known as *superposition*. However, these local structures have not been observed in real language models. In contrast, language models display rich structures like semantically clustered representations or ordered circles for the months of the year which are not predicted by current theories. In this work, we introduce Bag-of-Words Superposition (BOWS), a framework in which autoencoders (AEs) with a non-linearity are trained to compress sparse, binary bag-of-words vectors drawn from Internet-scale text. Our framework reveals that under restrictive bottlenecks, or when trained with weight decay, non-linear AEs linearly encode the low rank structure in the data, arranging feature representations according to their co-activation patterns. This *linear superposition* gives rise to structures like ordered circles and semantic clusters, similar to those observed in language models. Our findings suggest that the semantically meaningful structures observed in language models could arise driven by compression alone, without necessarily having a functional role beyond efficiently arranging feature representations.

1 INTRODUCTION

Despite the constant progress in Deep Learning (DL), our ability to trust DL models is limited by our lack of understanding of their inner workings. DL models, including Large Language Models (LLMs), are still considered ‘black boxes.’ The field of mechanistic interpretability (MI) aims to address these limitations by decomposing models into interpretable parts such as features or circuits, and understanding how these interact (Olah et al., 2020). A promising avenue within this field focuses on identifying meaningful “features” within model activations, using techniques like dictionary learning, particularly sparse autoencoders (SAEs) (Bricken et al., 2023; Templeton et al., 2024; Gao et al., 2025). These approaches have successfully uncovered interpretable units corresponding to semantic concepts, syntactic roles, or specific input patterns.

While methods for finding features continue to advance, our understanding of how these features arrange themselves geometrically within high-dimensional activation spaces remains limited. This problem was highlighted as one of the main open problems in MI (Sharkey et al., 2025), while Hindupur et al. (2025) highlighted the importance of understanding feature geometry to design better SAEs. A key challenge arises from the phenomenon of *superposition* (Elhage et al., 2022), where models represent more features than the number of dimensions of the activation space, forcing features to interfere with each other. Early theoretical explorations using simplified “toy models” (Elhage et al., 2022) suggested that, under superposition, features might optimally arrange themselves into local geometric structures, such as the vertices of regular polytopes or antipodal pairs (Figure 1 middle). These arrangements, combined with non-linearities like the Rectified Linear Unit (ReLU), could allow the model to filter out interference from features in the same local structure.

However, regular polytope structures have not been observed in standard LLM activations, instead, researchers have found evidence of different kinds of geometric organization, such as striking ordered

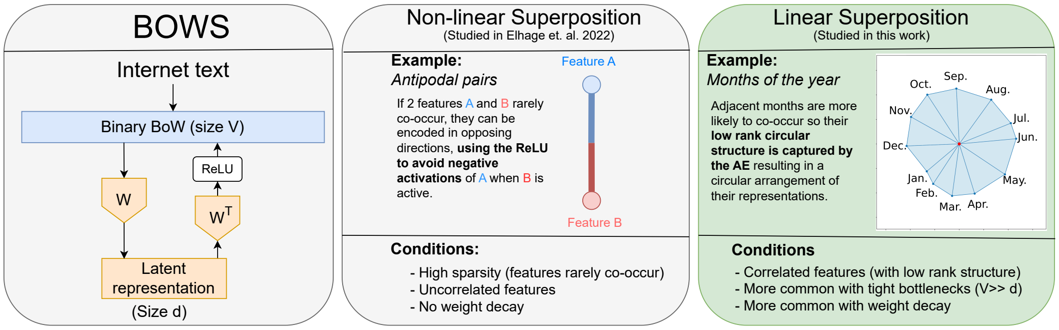

BOWS
Internet text
Binary BoW (size V)
W
ReLU
W^T
Latent representation
(Size d)
Non-linear Superposition
(Studied in Elhage et. al. 2022)
Example:
Antipodal pairs
If 2 features A and B rarely co-occur, they can be encoded in opposing directions, using the ReLU to avoid negative activations of A when B is active.
Feature A
Feature B
Conditions:
- High sparsity (features rarely co-occur)
- Uncorrelated features
- No weight decay
Linear Superposition
(Studied in this work)
Example:
Months of the year
Adjacent months are more likely to co-occur so their low rank circular structure is captured by the AE resulting in a circular arrangement of their representations.
Sep.
Oct.
Aug.
Nov.
Jul.
Jun.
Dec.
Jan.
May.
Feb.
Mar.
Apr.
Conditions
- Correlated features (with low rank structure)
- More common with tight bottlenecks (V>> d)
- More common with weight decay

Figure 1: **BOWS, our new framework to study superposition in realistic data (left) highlights the existence of two kinds of superposition, linear (middle) and non-linear (right).** Linear superposition appears under more realistic circumstances (correlated features, weight decay) explaining the prevalence of the phenomena caused by linear superposition in real language models.

circles of features (e.g., days of the week, months of the year) (Engels et al., 2025) and broader semantic clusters (Bricken et al., 2023; Templeton et al., 2024). Intriguingly, such circles and clusters resemble the global semantic organization first reported in distributional word embeddings (e.g., Word2Vec, GloVe), which arise from compressing word co-occurrence statistics (Mikolov et al., 2013b; Levy & Goldberg, 2014). This motivates exploring whether similar principles – structure arising as a byproduct of compressing data statistics – might also explain the geometric patterns observed within the internal representations of modern deep learning models.

To investigate these phenomena, we introduce Bag-of-Words Superposition (BOWS), a framework designed to investigate feature geometry in a controllable way with known ground-truth features. BOWS involves training a simple autoencoder (AE) – with a linear encoder and a linear decoder followed by a ReLU non-linearity (Figure 1 left) – to reconstruct sparse, binary bag-of-words vectors derived from large-scale internet text corpora. This setup is designed to mimic the behavior of a residual stream in a transformer-like model, which is often thought of as storing features in superposition without performing any computation (Elhage et al., 2021; Hänni et al., 2024) with the crucial difference that in BOWS we have access to the ground-truth features. Using BOWS reveals that when encoding features with realistic correlations in superposition, and particularly when using weight decay, the features are encoded in linear superposition, which gives rise to semantically meaningful structures rather than the previously studied regular polytopes.

Our main contributions are as follows:

- We design the BOWS framework as a controlled setting to study superposition in a controlled setting with realistic features.
- We introduce the distinction between non-linear and linear superposition. With the latter corresponding to non-linear autoencoders linearly encoding low-rank structure of the data, giving rise to semantically meaningful representations in the case of text data.
- We use BOWS to establish the prevalence of linear superposition in autoencoders with tight bottlenecks or trained with weight decay.
- We show that linear superposition in our BOWS setup reproduces and provides an explanation for the geometry of feature representations in real language models as a byproduct of compression.
- We introduce the distinction between presence-coding and value-coding features to explain the existence of structured representations in the absence of feature correlations (Section 5).

Our findings suggest that the feature geometry observed in real language models can be parsimoniously explained as the optimal arrangement of some features under linear superposition, without appealing to any functional relevance of these structures. We will make our implementation and experimental setup publicly available upon publication and it is currently available in the supplementary material.

2 Background

We introduce definitions of linear and non-linear superposition and present our setting for studying superposition under realistic data distributions. We define superposition for abstract features $\mathbf{f}$, then discuss the relevance of superposition when these features are properties of data under the linear representation hypothesis.

2.1 Definitions

Let $\mathcal{D}_\mathbf{f}$ be a distribution over vectors of features $\mathbf{f} = [f_1, \dots, f_d]^\top \in \mathbb{R}^d$, with $\text{Var}_{\mathcal{D}_\mathbf{f}}[f_i] > 0$ for all i. We consider linear encoders $W \in \mathbb{R}^{m \times d}$ with $m < d$, and representations $\mathbf{z} = \mathbf{Wf}$. Given a decoder $\psi : \mathbb{R}^m \to \mathbb{R}^d$, we define a score function

$$\mathcal{S}_{\mathcal{D}_\mathbf{f}}(\mathbf{W}, \psi) = 1 - \frac{1}{d} \sum_{i=1}^{d} \frac{\mathbb{E}_{\mathcal{D}_\mathbf{f}}\left[(f_i - \psi(\mathbf{Wf})_i)^2\right]}{\text{Var}_{\mathcal{D}_\mathbf{f}}[f_i]}, \tag{1}$$

corresponding to the mean coefficient of determination (R^2) across coordinates.

Definition 1 (Superposition). Let $\mathbf{W} \in \mathbb{R}^{m \times d}$ with $m < d$. We say that $\mathbf{W}$ **encodes the features of $\mathcal{D}_\mathbf{f}$ in superposition** if there exists a decoder $\psi : \mathbb{R}^m \to \mathbb{R}^d$ such that $\mathcal{S}_{\mathcal{D}_\mathbf{f}}(\mathbf{W}, \psi) \geq 1 - \varepsilon$ for a chosen tolerance $\varepsilon \geq 0$.

Definition 2 (Linear Superposition). We say that $\mathbf{W}$ encodes the features of $\mathcal{D}_\mathbf{f}$ in *linear superposition* if there exists a **linear** decoder $\psi_{\text{lin}} : \mathbb{R}^m \to \mathbb{R}^d$ such that $\mathcal{S}_{\mathcal{D}_\mathbf{f}}(\mathbf{W}, \psi_{\text{lin}}) \geq 1 - \varepsilon$.

Definition 3 (Non-linear Superposition). We say that $\mathbf{W}$ encodes the features of $\mathcal{D}_\mathbf{f}$ in *non-linear superposition* if

1. there exists a non-linear decoder $\psi : \mathbb{R}^m \to \mathbb{R}^d$ with $\mathcal{S}_{\mathcal{D}_\mathbf{f}}(\mathbf{W}, \psi) \geq 1 - \varepsilon$, and
2. for every linear decoder $\psi_{\text{lin}} : \mathbb{R}^m \to \mathbb{R}^d$, we have $\mathcal{S}_{\mathcal{D}_\mathbf{f}}(\mathbf{W}, \psi_{\text{lin}}) < 1 - \varepsilon$.

Superposition in deep learning models. The definitions above apply to any features $\mathbf{f}$. However, superposition becomes central to mechanistic interpretability under the *linear representation hypothesis* (LRH). The LRH reflects the empirical finding that high-level concepts such as language (Gurnee et al., 2023), entity attributes, or specific landmarks (Templeton et al., 2024) are often linearly represented in model activations. Concretely, let $\mathbf{x} \in \mathcal{X}$ denote a data sample (e.g., an image or text) and let $\rho_1, \dots, \rho_d : \mathcal{X} \to \mathbb{R}$ be interpretable properties (e.g., "is written in French", "contains a dog"). These induce a feature vector $\mathbf{f}(\mathbf{x}) = [\rho_1(\mathbf{x}), \dots, \rho_d(\mathbf{x})]^\top$.

Definition 4 (Linear Representation Hypothesis). Let $\mathbf{x} \in \mathcal{X}$ be a data sample and $h(\mathbf{x}) \in \mathbb{R}^m$ be a neural network layer's activation. The network *linearly represents* properties $\{\rho_j\}_{j=1}^d$ with $\rho_j : \mathcal{X} \to \mathbb{R}$ as directions $\{\mathbf{w}_j\}_{j=1}^d$ if:

$$h(\mathbf{x}) \approx \sum_{j=1}^{d} \rho_j(\mathbf{x}) \mathbf{w}_j. \tag{2}$$

A model that linearly represents more concepts than its hidden dimension must encode them in superposition for downstream use. Under the LRH, understanding how these concepts are geometrically arranged is thus a key challenge for mechanistic interpretability.

2.2 BOWS: Realistic data in superposition

Studying superposition in the hidden representations of deep learning models relies on postulating some properties of the data $\rho(\mathbf{x})$ which are linearly represented. However, in most cases we do not have access to a ground truth for these features or their representations. We now introduce Bag-of-Words Superposition (BOWS) as a setup to study superposition of features drawn from a realistic distribution that goes beyond the iid. and pairwise correlation cases studied in Elhage et al. (2022) and introduces low rank structure in the covariance of the data.

Dataset. Let $\mathcal{C}$ be a corpus of text segmented into *records* (lines or paragraphs). After word-level tokenisation, we construct a vocabulary of the V most frequent words, discarding common English stop-words and prepositions. This vocabulary includes words such as *sun*, *code*, and *January* which often correspond to linear features in sparse autoencoders trained on language data (Engels et al., 2025; Bricken et al., 2023). Each record is then encoded as a binary bag-of-words vector $\mathbf{x} \in \{0, 1\}^V$ whose j-th component is 1 iff the j-th vocabulary word appears in the record.

We choose a *context size* $c \in \mathbb{N}$. For every contiguous block of c records we take the element-wise logical OR of their individual vectors, obtaining a single sample. The resulting dataset is

$$\mathcal{D} \;=\; \{\mathbf{x}_i\}_{i=1}^{N}, \qquad \mathbf{x}_i \in \{0,1\}^V, \tag{3}$$

where N is the number of c-record chunks in the corpus.

Experiments in the main text use the WikiText-103 corpus (Merity et al., 2017). With $V = 10{,}000$ and $c = 10$ we obtain $N = 1{,}801{,}255$ training examples. We refer to this pre-processed collection as WikiText-BOWS. We include replication of the main results using OpenWebText in Appendix D.

Autoencoder. We use the autoencoder setup for superposition introduced in Elhage et al. (2022), consisting of an encoder with weights $\mathbf{W} \in \mathbb{R}^{m \times V}$ and bias $\mathbf{b} \in \mathbb{R}^V$, where the input $\mathbf{f} \in \mathbb{R}^V$ is reconstructed using a ReLU AE with loss:

$$\mathcal{L}_{\mathrm{ReLU-AE}}(\mathbf{x}, \mathbf{W}, \mathbf{b}) = ||\mathbf{f} - \mathrm{ReLU}(\mathbf{W}^T\mathbf{W}\mathbf{f} + \mathbf{b})||_2^2 \tag{4}$$

We also use a Linear AE as a baseline with loss:

$$\mathcal{L}_{\text{Linear-AE}}(\mathbf{f}, \mathbf{W}, \mathbf{b}) = ||\mathbf{f} - (\mathbf{W}^T\mathbf{W}\mathbf{f} + \mathbf{b})||_2^2 \tag{5}$$

3 LINEAR SUPERPOSITION IN NON-LINEAR AES

We now study how data covariance structure and optimization constraints shape the solutions learned by linear and non-linear autoencoders. While linear AEs are restricted to *linear superposition*, non-linear models such as ReLU-AEs can leverage both linear and non-linear superposition. We characterize when each regime emerges.

3.1 SUPERPOSITION AND INTERFERENCE

Superposition requires encoding more features than dimensions, implying non-orthogonal representations and thus *interference* between features. For tied-weight AEs with weight matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$ and columns $\{\mathbf{w}_i\}_{i=1}^{d}$, the reconstruction of feature f_i decomposes as:

$$\hat{f}_i = \sigma\Big(\underbrace{\|\mathbf{w}_i\|^2 f_i}_{\text{Signal}} + \underbrace{\sum_{j \neq i} \langle \mathbf{w}_i, \mathbf{w}_j \rangle f_j}_{\text{Interference } \mathcal{I}_i} + b_i\Big). \tag{6}$$

In the linear case, the optimal bias satisfies $b^\star = (I - \mathbf{W}^\top\mathbf{W})\mathbb{E}[f]$, so the autoencoder effectively operates on centered features $f - \mathbb{E}[f]$. We therefore analyse the centered setting with $b = 0$ when $\sigma = id$ without loss of generality.

Let $\mathbf{\Sigma} = \mathbb{E}[f f^\top]$ denote the feature covariance. Interference from superposition is typically viewed as noise that must be filtered out non-linearly (Elhage et al., 2022). We show this is not always the case: when $\mathbf{\Sigma}$ is approximately low rank, interference can be *harnessed* rather than filtered, aligning constructively with the signal.[1]

Non-linear superposition ($\mathbf{\Sigma} \approx I$, sparse features). When features are independent and sparse, $\mathcal{I}_i$ is indeed unstructured noise uncorrelated with f_i. Accurate reconstruction requires filtering this interference via a non-linearity (e.g. $\sigma = \mathrm{ReLU}$) and negative bias such that $\mathbb{E}[\mathrm{ReLU}(\mathcal{I}_i + b_i)] \approx 0$, while maintaining $\|\mathbf{w}_i\|^2 \approx 1$ to preserve the signal. This yields $\|\mathbf{W}\|_F^2 \approx d$.

Linear superposition ($\mathrm{rank}(\mathbf{\Sigma}) \leq m$). For linear AEs ($\sigma = id$), the optimal map $\mathbf{P} = \mathbf{W}^\top\mathbf{W}$ is the orthogonal projector onto the top-m principal components of $\mathbf{\Sigma}$ (Baldi & Hornik, 1989; Jolliffe, 2002). Defining the reconstruction residual $\varepsilon_i = f_i - \hat{f}_i$, we can rearrange Equation (6) to isolate interference:

$$\mathcal{I}_i = (1 - \|\mathbf{w}_i\|^2) f_i - \varepsilon_i. \tag{7}$$

[1] Elhage et al. (2022) also consider pairwise correlated features, but in their setup, all principal components carry significant variance and PCA collapses correlated pairs onto indistinguishable points. However, when Σ is low-rank or has fast spectral decay, fewer principal components suffice making linear superposition viable.

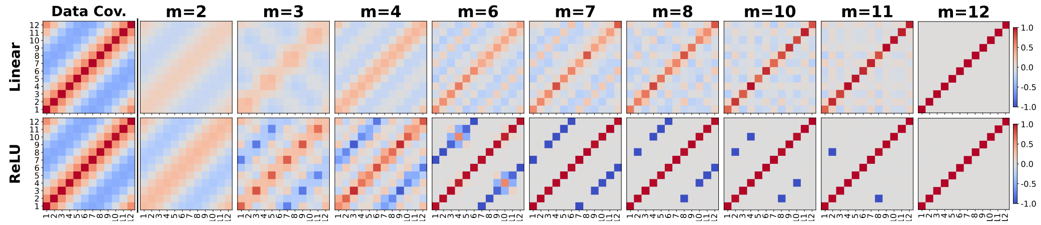


Figure 2: **Autoencoding synthetic correlated features reveals different superposition regimes.** Weight inner products ($\mathbf{W}^\top\mathbf{W}$) at convergence for AEs encoding $d = 12$ features with cyclic covariance, varying latent size m. **Top (Linear AE):** Captures top-m principal components; at $m = 2$, represents all 12 features via the circular structure. **Bottom (ReLU AE):** Matches linear AE for small m (linear regime), but forms antipodal pairs for larger m (non-linear regime) to exploit ReLU for interference filtering.

When $\text{rank}(\boldsymbol{\Sigma}) \leq m$, the data lies in the principal subspace, so $\varepsilon = 0$ and $\mathcal{I}_i = (1 - P_{ii})f_i$, meaning interference is proportional to the signal. This arises because $P_{ij} = \langle \mathbf{w}_i, \mathbf{w}_j \rangle$ reflects the correlation between features i and j within the principal subspace. Each f_j contributes to the reconstruction of f_i in proportion to their shared variance. This reframing of PCA through the lens of superposition clarifies why interference need not be filtered: for correlated data, the optimal linear solution arranges representations so that interference reinforces the signal. In practice, P is learned from finite data and reflects the sample covariance; for test samples whose correlations deviate from the training distribution, interference will be imperfectly aligned with the signal.

The case of real data($\boldsymbol{\Sigma}$ approximately low rank). In real-world data, such as the ones we consider, the covariance is often well approximated by a few principal components (Deerwester et al., 1990; Blei et al., 2003; Udell & Townsend, 2019), rendering $\boldsymbol{\Sigma}$ approximately low-rank. In this regime, interference acts as signal contaminated by a residual ε_i, corresponding to variance orthogonal to the top-m eigenvectors: $\|\varepsilon\|^2 = \sum_{k>m} \lambda_k(\boldsymbol{\Sigma}) \geq \min_{\text{rank}(\widehat{\boldsymbol{\Sigma}})\leq m} \|\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\|_F$ (Eckart & Young, 1936). In other words, the interference concentrates along low energy directions, lower bounding the Frobenius-optimal truncation error.

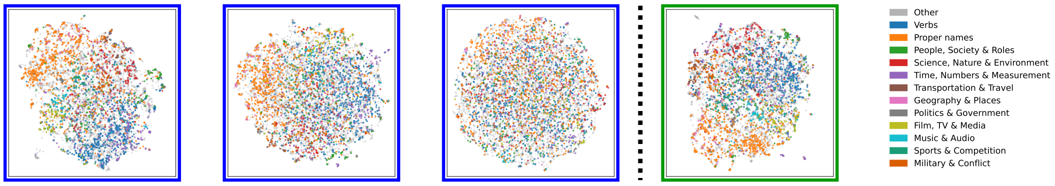
When do non-linear models use linear superposition? Considering models trained on real data using weight decay. Non-linear superposition requires $\|\mathbf{W}\|_F^2 \approx d$, while linear superposition with an m-dimensional projector satisfies $\|\mathbf{W}\|_F^2 = \sum_k \lambda_k(P) = m < d$. Therefore, *weight decay combined with tight bottlenecks* ($m \ll d$) biases even non-linear models toward linear superposition, where a linear decoder $\psi_{\text{lin}} = \mathbf{W}^\top$ suffices (Definition 2). Since ReLU-AEs lack closed-form solutions, we verify this prediction empirically.

3.2 Evidence of linear superposition in non-linear AEs

We start our empirical analysis with a simplified setting where two AEs of latent dimension m with and without a ReLU in the decoder are trained on 12 dimensional data with a cyclic covariance structure (Figure 2 left). Details for the data generation process are provided in Appendix A. Figure 2 (top row, Linear) shows the baseline learned by the linear AE, which learns the projection onto the principal subspace (Baldi & Hornik, 1989). We then study the emergence of linear superposition in non-linear autoencoders, specifically looking at autoencoders with a ReLU in the decoder of the kind described in Section 2.2.

Linear superposition. When the bottleneck is very tight ($m \ll d$), the ReLU AE recovers the circular structure dictated by the top principal components Figure 2 (bottom row, ReLU, $m < 6$). These are examples of a non-linear AE leveraging linear superposition.

Non-linear superposition. Figure 2 (bottom row, ReLU, $m \geq 6$) shows that as m increases, the ReLU AE abandons the circular PCA structure and instead represents features as *antipodal pairs*. This specific geometry is one of the cases studied by Elhage et al. (2022), whereby features are placed in anti-correlated pairs such that activating one feature negatively activates its antipodal partner, with negative interference zeroed out by the ReLU as an example of non-linear superposition.



(a) $m = 200$ $wd = 0$ (b) $m = 400$ $wd = 0$ (c) $m = 600$ $wd = 0$ (d) $m = 600$ $wd = 4$ (e) Semantic categories

Figure 3: **Linear superposition appears in ReLU AEs which have small latent sizes (a) or are trained with weight decay (d), giving rise to semantic clusters.** UMAP projections of word embeddings from AEs of different latent dimensions (m) and weight decay values (wd). Points are colored by semantic category (e).

Structure disappears as features become orthogonal. As the latent size approaches the input size, the autoencoder weights converge to the identity, representing each feature orthogonally for a perfect reconstruction. As these models ($m = 12$) approach a perfect reconstruction of the data, the circular covariance structure is no longer reflected in the weights, and any notion of superposition is lost.

4 Linear superposition explains feature geometry in real data

In Section 3, we showed that ReLU AEs can leverage linear superposition particularly when trained with weight decay on correlated data, linearly capturing the covariance structure in the weights and leading to constructive interference that does not need to be filtered out. In this section we extend this analysis to our main WikiText BOWS setting.

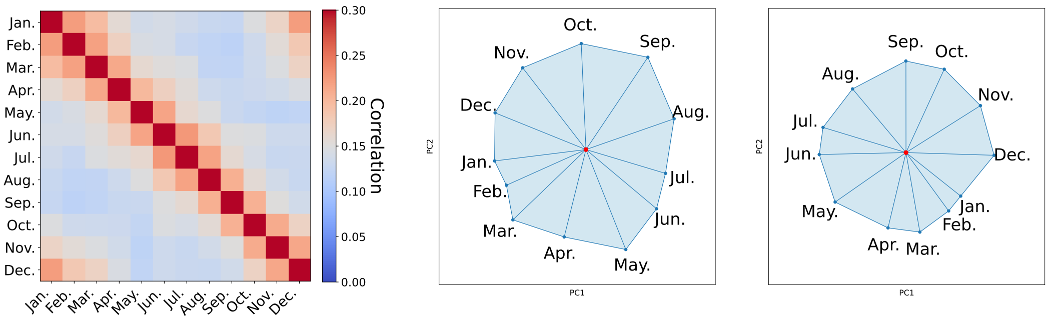
Semantic clustering under linear superposition. One prominent observation in studies interpreting LLM activations is that learned features often form clusters based on semantic relatedness leading to *anisotropic superposition*, where features are no longer arranged to minimize pairwise dot products (Bricken et al., 2023; Templeton et al., 2024). While previously unexplained, these kinds of structures are to be expected if LLMs leverage linear superposition to capture low rank structure in the data as described in Section 3.

In Figure 3 we show UMAP (McInnes et al., 2018) projections of the word-embeddings (columns of $\mathbf{W}$) learned by a ReLU AE trained on WikiText-BOWS ($V = 10,000$) with varying latent dimensions m. In panel (a), where compression is high ($m = 200$), we observe distinct clusters corresponding to semantic categories (e.g., verbs, proper names, sports). While this clustering disappears as the latent size increases to $m = 600$ (Figure 3c), introducing weight decay during training recovers this structure even at larger latent sizes (Figure 3d). These results suggest linear superposition as a straightforward mechanism for the semantic clustering observed both in BOWS and in the internal representations of LLMs which are trained with weight decay.

Cyclical structures inherited from data statistics. Another notable observation of feature geometry in real models is that of circular structures appearing in the principal components of feature embeddings. This has been observed for concepts like the months of the year or the days of the week (Engels et al., 2025).

Consider the features corresponding to the twelve months of the year. In Figure 4a, we show that there is a cyclical correlation structure between these features in WikiText. For example January co-occurs more often with February and December than with August. This cyclical correlation drives the leading principal components of the month activations, producing a circle in the 2D PCA plot Figure 4b. Notably, the learned latent representations reflect the same geometry (Figure 4c). For a fully linear AE this would be expected since its weights span the PCA subspace (Baldi & Hornik, 1989) (more detailed argument in Appendix C). However, our non-linear AE shows the same pattern, supporting the existence of linear superposition in non-linear AEs.

Compression alone being sufficient to give rise to this circular structure in the PCA of the weights suggests that it may not be actively constructed by language models for a specific non-linear function (e.g. modular addition (Engels et al., 2025)), but rather passively *inherited* from the statistical structure of the input data when subjected to dimensionality reduction.



(a) Data correlations

(b) PCA of the data

(c) PCA of the weights

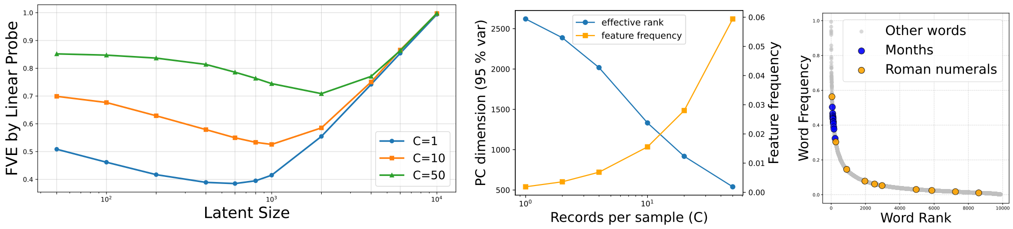
Figure 4: **Circular representation of months arises from data covariance via PCA. (Left)** Empirical correlation matrix of month words in the WikiText-103 BOWS dataset, showing cyclic correlations. **(Middle)** PCA applied directly to the 12 month dimensions of the WikiText-103 BOWS data vectors, projected onto the top 2 PCs, reveals a circle. **(Right)** PCA applied to the 12 learned encoder features (W columns) for months from a ReLU AE trained on WikiText-BOWS ($V = 10k$) with ($m = 1000$), projected onto their top 2 PCs, also recovers the circular structure. This suggests the AE inherits the structure from the data via PCA-like compression.

Characteristics of realistic data and their impact on linear superposition. A key characteristic of the BOWS setup is that the correlations in realistic data introduce low rank structure, making linear superposition possible. Figure 5a shows that the amount of linear superposition, measured by the FVE using a linear probe, increases significantly as we increase the context size (details for the FVE methodology are provided in Section 2.2). Figure 5b also shows that as the context size increases, the number of principal components needed to explain 95% of the data decreases. This shows that a larger context leads to lower sparsity and stronger correlations making representing principal components more beneficial, and aligning with the observation by Elhage et al. (2022) that non-linear superposition requires sparsity. These findings suggest a possible explanation for why residual streams of LLMs display similar patterns to those observed in AEs performing linear superposition (Templeton et al., 2024), as LLMs often have very large context windows which can increase correlations in the data.

While the features in our setup in Section 3 all occur with the same frequency leading to distinct linear and non-linear superposition regimes, word occurrences on real data are known to follow a power-law distribution (Zipf, 1949; Clauset et al., 2009). Therefore, words have a wide range of frequencies Figure 5c which can lead to a hybrid regime where some words reflect their covariance structure while others are in non-linear superposition or fully orthogonal to each-other. In Figure 6a we show an example of this by isolating two groups of words: the months of the year, and the first 10 roman numerals. We isolate the weights $\mathbf{W}_{\text{months}}$ and $\mathbf{W}_{\text{roman}}$ and take the off diagonal Frobenius norm of the matrices $\mathbf{W}_{\text{months}}^T \mathbf{W}_{\text{months}}$ and $\mathbf{W}_{\text{roman}}^T \mathbf{W}_{\text{roman}}$ to measure how much they interfere with each-other. An off-diagonal Frobenius norm of 0 means all the months or roman numerals are orthogonal to each-other. Figure 6 shows that the weights of both the months and the roman numerals appear in order, reflecting their covariance structure when the latent size is small. As the latent size increases, the structure in $\mathbf{W}_{\text{months}}$ disappears first as the off-diagonal Frobenius norm approaches 0. This example showcases that, while in the simple case described in Section 3 we observe distinct regimes (linear and non-linear), in realistic data with features of different frequencies, groups of features can be in different regimes in the same model.

5 VALUE-CODING FEATURES: ANOTHER CAUSE OF FEATURE GEOMETRY

While the BOWS framework can replicate the kind of semantic structure observed in the hidden representations of language models, there are some examples, like the circles that appear in models performing modular addition (Power et al., 2022; Nanda et al., 2023), which appear in the absence of correlations in the data. To explain this kind of structure, we introduce the distinction between *value-coding* and *presence-coding* features and explain how value-coding features can give rise to apparent structures in features that are not actually represented in superposition.



(a) (b) (c)

Figure 5: **Bigger context size** leads to **stronger correlations**, thus **linear superposition**. (**Left**) Increasing the amount of text encoded by each sample (C) increases the fraction of the variance of the ReLU AE reconstructions that can be explained from its latent space by a linear probe. (**Middle**) Increasing C also decreases the number of PCs needed to account for 95% of the variance while increasing the average number of active features per sample. (**Right**) Word frequencies follow a power-law distribution (Zipf, 1949) with words like the months being more common than roman numerals.

Presence-coding features. We say that a representation $h(x) \in \mathbb{R}^d$ contains a *presence-coding feature* if some binary or categorical variable $y(x)$ (e.g. "this token is the word *cat*") is recoverable by a linear classifier. Formally, there exist weights $\{w_k, b_k\}$ such that $\hat{y}(x) = \arg\max_k(w_k^\top h(x) + b_k)$ predicts $y(x)$ with low error. Presence-coding features thus behave as detectors for discrete properties, and different values of y are treated as separate classes without requiring any particular geometric relation between them in representation space *a priori*. For presence-coding features structured representations are contingent on correlations in the data and capacity constraints that lead the features to be represented in linear superposition.

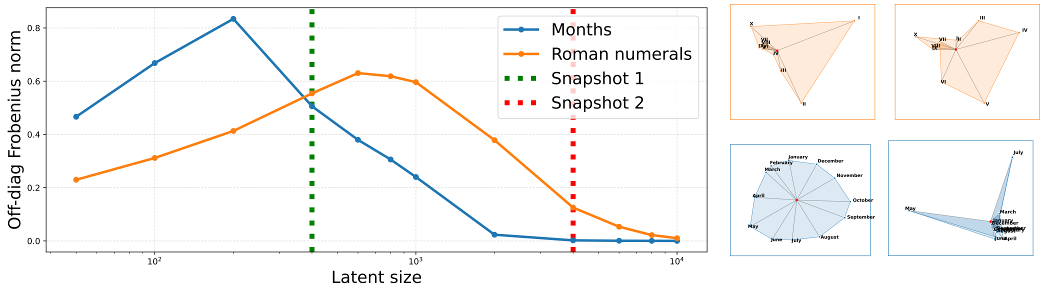
Value-coding features. In contrast, we say that a representation $h(x)$ contains a *value-coding feature* if a real-valued variable $v(x) \in \mathbb{R}$ (e.g. an angle, a coordinate, or a continuous latent factor) is linearly decodable. That is, there exist $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $\hat{v}(x) = w^\top h(x) + b$ approximates $v(x)$ with low error. A collection of such value-coding features $v_1(x), \ldots, v_k(x)$ defines a low-dimensional *value space* $\mathbb{R}^k$; plotting examples in this space can reveal semantically meaningful structures (for instance, a 2D map from (x, y) coordinates, or a circle from $(\sin\theta, \cos\theta)$ in modular addition). Crucially, these structures are fully accounted for by the existence of linear value codes for the underlying variables and therefore exist even in the absence of superposition.

Empirical evidence for value-coding features To exemplify this, we look at a simplified modular addition setup similar to that in Nanda et al. (2023), as well as a relative map position setup inspired by Gurnee & Tegmark (2024). In the latter, we take the top 1,000 most populated cities in the US and calculate their relative positions in terms of quadrants (e.g. Seattle is north-west of Denver). The cities are embedded separately, fed through a 1-hidden layer ReLU MLP (details in Appendix A). This latter dataset is designed to incentivize the model to learn 2 value-coding features encoding the coordinates of each city since relative positions of the cities can easily be calculated by subtracting coordinates. For both of these tasks, each integer or city pair only appears once across the train and validation sets so no pair of cities or integers are correlated and we do not expect any kind of low rank structure coming from linear superposition.

Table 1: Value-coding (VC) ablations on two datasets. VC^+ keeps the VC subspace and zeros its orthogonal complement; VC^- ablates VC coordinates.

Condition	MAP		Key-freq	
	Loss ↓	Acc. (%) ↑	Loss ↓	Acc. (%) ↑
Baseline	0.0536	97.94	0.0001	100.00
VC^+	0.2879	93.16	0.1649	93.99
VC^-	6.3943	22.43	11.7464	3.11

We validate that the cardinal direction model is learning value coding features for the coordinates by training a linear probe to predict the coordinates of a subset of the cities from their embeddings. We then use this probe to predict the coordinates of some held-out cities. The result is an average R^2 validation score of 0.98 and a correct arrangement of the held out cities on the US map when projected onto the directions identified by the linear probes (Figure 7). Similarly, we validate that the relevant sine and cosine values are linearly represented by projecting the representations onto the corresponding Fourier components in the case of modular addition (details in Appendix A).



(a) Sweep over latent sizes (b) Snapshot 1 (c) Snapshot 2

Figure 6: **Different feature structures disappear at different latent sizes.** As the latent size of the AE increases, the structure in the weights appears and disappears at different times for different groups of figures (**a**). We look at 2D PCA plots of the months and roman numeral embeddings at two key latent sizes. In (**b**) we see a case where both groups of features appear in order with some structure, while in (**c**) we show a case where the representations of the months have already become orthogonal while the roman numerals are still represented in an ordered structure.

We have shown two examples where neural networks need linearly represent values like coordinates or trigonometric functions, in order to perform computations with them. We have then shown in Figure 7 that in doing this, models implicitly create structures like circles or maps when we project the data onto these value-coding features.

Ablating the subspace orthogonal to the value-coding features. Having isolated the value coding features both in the modular addition and map datasets, we ablate the subspace of the embedding space that is orthogonal to these value-coding features. In both cases, the ablation results show that most of the test accuracy is preserved if we replace over 90% of the dimensions with their mean (Table 1). Conversely performance breaks down if we remove only the value coding features. This serves as further evidence that these value coding features are the units of computation that the model is using to perform these tasks (replicating the results from Nanda et al. (2023) in the modular case).

Distinguishing feature geometry and feature manifolds. At first glance, feature manifolds such as those in Figure 7 may appear deceptively similar to geometric arrangements like Figure 4. Yet, our findings offer a principled way to distinguish the two. When features exhibit a recognizable structure, we can ask whether that structure reflects genuine co-activation patterns. In Figure 4, for instance, the latent arrangement clearly aligns with co-activations among month-related features, pointing to a superposition-based representation. In contrast, the patterns in Figure 7 emerge despite inputs being uncorrelated, suggesting the model has instead learned value-coding features through task-driven projections. Disentangling these phenomena in more complex, real-world scenarios remains a compelling direction for future research.

6 Related work

Superposition. Initial works in MI studied interpretable monosematic neurons in DL models Olah et al. (2020); Cammarata et al. (2020) but faced challenges in interpreting polysemantic neurons which activate for seemingly unrelated concepts. Elhage et al. (2022) introduced superposition as an explanation for neuron polysemanticity. This view of DL models inspired further studies (Scherlis et al., 2025) and sparse dictionary learning approaches like sparse autoencoders to decompose model activations into an overcomplete basis of linear features (Gurnee et al., 2023; Huben et al., 2024; Bricken et al., 2023). This approach has successfully been scaled to frontier language models and multimodal models by Gao et al. (2025) and Templeton et al. (2024).

Feature geometry. Park et al. (2024) proposed a formalization of the LRH and proposed an inner product that preserves language structure. Park et al. (2025) studied how features with hierarchical relations are encoded in language models while they show that categorical features which form polytopes, we note that these are different from *regular* polytopes posited by Elhage et al. (2022). Lee et al. (2025) studied geometric similarities in token embeddings of different language models, while Zhao et al. (2024) studied the kind of structure that emerges in the representations of models trained on next token prediction. Li et al. (2025) showed that language models represent integers in a helix structure to perform modular addition echoing the results from Nanda et al. (2023) and

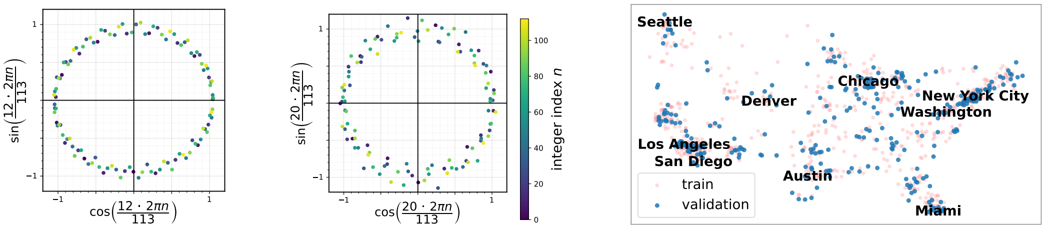


Figure 7: In the embeddings of a model performing modular addition, the circular structure is isolated by projecting onto directions corresponding to sine and cosine values (**left**). In the case of the map directions dataset, two linear probes reconstruct the city coordinates, reconstructing their positions on the map for both train and validation samples. (**right**).

Liu et al. (2022) on transformers trained for modular addition. Gurnee & Tegmark (2024) showed that longitude and latitude as well as a notion of time, are encoded as linear features in language models. The main results highlighted in this paper are circular structures formed by features and semantic feature clusters, described in Engels et al. (2025) and Bricken et al. (2023) respectively. These structures have also been studied as feature manifolds in Modell et al. (2025), and Hindupur et al. (2025) highlighted the importance of understanding feature geometry when designing SAEs. These findings sparked a discussion around the potential limitations of SDL approaches and the LRH suggested by this non-linearly encoded semantic information (Sharkey et al., 2025).

Structure in word representations. Classic work on distributional semantics and word embeddings (e.g., Word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014)) demonstrated that training simple models on large text corpora leads to vector spaces where geometric relationships capture surprisingly sophisticated semantic and syntactic relationships. Levy & Goldberg (2014) showed that methods like Word2Vec with negative sampling implicitly factorize the Pointwise Mutual Information (PMI) matrix shifted by a constant, while others show connections to PCA or SVD on co-occurrence counts or PMI (Allen & Hospedales, 2019).

7 Discussion & Conclusion

Summary of findings. In this work, we introduce BOWS to highlight the existence of two kinds of superposition, linear and non-linear. We argue that features in linear superposition inherit their structure from their covariance matrix, explaining previously observed, semantically rich structures in language models (Bricken et al., 2023; Engels et al., 2025). We show this on synthetic data(Section 3) as well as in realistic internet data (Section 4) where semantic clusters and circles appear as a byproduct of linear superposition. We show linear superposition is particularly prevalent in models trained with weight decay, with potential implications about the kind of superposition we should expect to observe in real language models. Finally, we make the distinction between presence-coding and value-coding features and show how value-coding features can parametrize feature manifolds, leading to a different kind of structure Section 5.

Limitations and future work. The BOWS framework, while simple, gives rise to many kinds of interesting behavior, leaving ample room for future work. As discussed in Section 4, we expect real models to exhibit both linear and non-linear superposition, leaving room to explore how linear and non-linear superposition can complement each-other. Additionally, while we show that structured representations do not require a functional role beyond efficient compression, this does not need to be a dichotomy as this kind of structure could be both functional and efficient.

Our results suggest the rank of the data, weight decay and bottleneck size as particularly relevant variables, but a more complete mathematical characterization of the relationship between linear and non-linear superposition is an important avenue for future work. Avenues for future work also include studying BOWS setups with untied encoder and decoder weights as well as using BOWS as a setup for SAE evaluation in which we know the arrangement of realistic ground truth features, addressing one of the main limitations in current SAE evaluations.

LLM USAGE STATEMENT

This work used LLMs to assist in literature search, suggest alternative phrasing and help with figure formatting. We also used an LLM to sort 4000 words into semantic categories which were then inspected and refined by hand, as discussed in Appendix A.

REPRODUCIBILITY STATEMENT

We include code to reproduce the main results of this paper (including Figure 2, Figure 3 and Figure 4) in the supplementary material. Details about the BOWS setup are given in Section 2.2 and further details about our experiments are provided in appendix A.

REFERENCES

Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 223–231. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/allen19a.html.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90014-2. URL https://www.sciencedirect.com/science/article/pii/0893608089900142.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

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

Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 2020. doi: 10.23915/distill.00024.003. https://distill.pub/2020/circuits/curve-detectors.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. doi: 10.1137/070710111.

Michael A. A. Cox and Trevor F. Cox. *Multidimensional Scaling*, pp. 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41 (6):391–407, 1990.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

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. https://transformer-circuits.pub/2022/toy_model/index.html.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

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

GeoNames. GeoNames cities1000 dataset. https://download.geonames.org/export/dump/, 2025. Accessed: 2025-05-19.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

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

Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=OcVJP8kClR.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry, 2025. URL https://arxiv.org/abs/2503.01822.

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

Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

J. Dennis Lawrence. *A Catalog of Special Plane Curves*. Dover, 1972.

Andrew Lee, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings. In *Second Workshop on Representational Alignment at ICLR 2025*, 2025. URL https://openreview.net/forum?id=F2IYiG0RLf.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/b78666971ceae55a8e87efb7cbfd9ad4-Paper.pdf.

Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4), 2025. ISSN 1099-4300. doi: 10.3390/e27040344. URL https://www.mdpi.com/1099-4300/27/4/344.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=6at6rB3IZm.

R. Marques, C. Bouville, M. Ribardière, L.-P. Santos, and K. Bouatouch. Spherical fibonacci point sets for illumination integrals. *Computer Graphics Forum*, 32(4):134–143, 2013.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090/.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models, 2025. URL https://arxiv.org/abs/2505.18235.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

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.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bVTM2QKYuA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

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

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

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.

Ozan Tuncer, Vitus J Leung, and Ayse K Coskun. Pacmap: Topology mapping of unstructured communication patterns onto non-contiguous allocations. In *Proceedings of the 29th ACM on International Conference on Supercomputing*, pp. 37–46, 2015.

Madeleine Udell and Alex Townsend. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1):144–160, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=qyilOnIRHI.

George Kingsley Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA, 1949.

APPENDIX

A IMPLEMENTATION DETAILS

WikiText-BOWS. All the models trained in the WikiText BOWS setup use a cosine annealing scheduler with a starting learning rate of $1e-3$ and are trained for 20 epochs with a batch size of 1024.

Synthetic "Months" dataset. Each document is a 12-bit vector $x \in \{0,1\}^{12}$ whose entries stand for the calendar months. One sample is generated as follows.

1. *Latent month angle.* Pick a discrete month $m \in \{0,\dots,11\}$ (uniformly or by cycling) and add Gaussian blur:
$$\theta = 2\pi m/12 \;+\; \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,\sigma_\theta^2).$$
2. *Embed on the unit circle.* $z = \big[\cos\theta,\ \sin\theta\big]^\top \in \mathbb{R}^2$.
3. *Project onto month directions.* Let
$$W = \Big[(\cos\tfrac{2\pi k}{12},\, \sin\tfrac{2\pi k}{12})\Big]_{k=0}^{11} \in \mathbb{R}^{12\times 2},$$
whose k-th row corresponds to month k. Compute log-odds $\ell_k = \beta\, W_k z + b$, where $b < 0$ fixes the global sparsity and $\beta > 0$ controls sharpness.
4. *Binary activations.* Draw the bits independently:
$$x_k \;\sim\; \text{Bernoulli}\big(\sigma(\ell_k)\big), \qquad \sigma(u) = \tfrac{1}{1+e^{-u}}, \qquad k = 1,\dots,12.$$

With σ_θ=0 and large β the code is nearly one-hot; decreasing β or increasing σ_θ mixes neighboring months, producing a rank-2 correlation structure that is analytically tractable yet retains the extreme sparsity of real bag-of-words data.

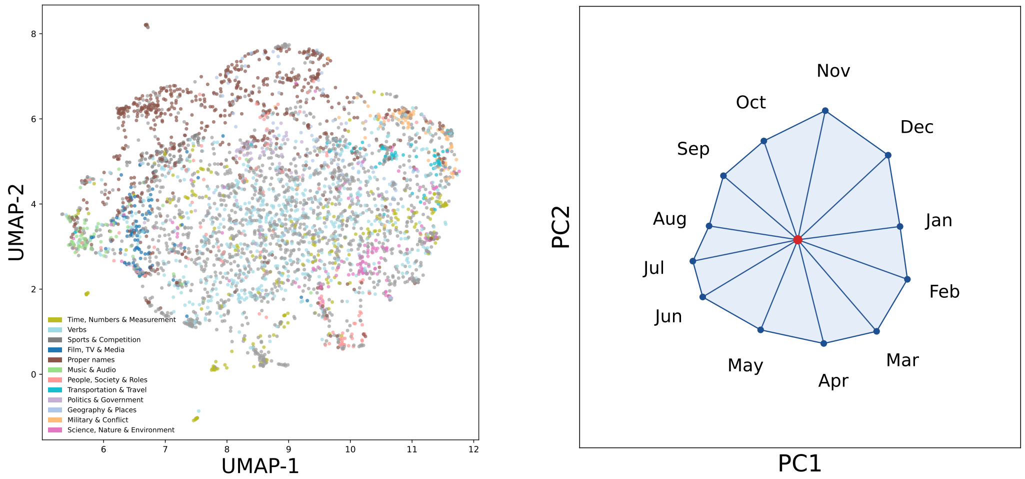
Modular addition. Let $a, b \in 0,\dots,112$ make an input pair of integers with the task being addition modulo 113. We learn a shared embedding matrix $E \in \mathbb{R}^{113\times 100}$ that maps each integer to a 100-dimensional vector. For a given pair, we look up $E[a]$ and $E[b]$, concatenate them into a 200-dimensional feature vector, and feed this through a three-hidden-layer MLP. Each hidden layer has 200 ReLU neurons. The output layer produces logits over the 113 possible sums. We train the entire model end-to-end via cross-entropy loss using the AdamW optimizer with weight decay set to 4.

Relative map positions. From the Cities1000 Dataset (GeoNames, 2025), we take the top 1,000 most populated cities in the US and sample two subsets of city pairs out of the $1{,}000{,}000$ possible pairs. For each pair of cities a, b, the task is to predict their relative position on the US map out of eight possible classes (North, South, East, West, North–East, North–West, South–East, South–West). We learn a 200-dimensional embedding matrix $E \in \mathbb{R}^{1000\times 200}$ to map each city to a 50-dimensional vector; for each pair (a, b), we concatenate their embeddings $E[a]$ and $E[b]$ into a 100-dimensional feature vector, which is then fed through a single hidden-layer MLP with 200 ReLU units. The MLP's output layer produces logits over the eight classes, and the entire model is trained end-to-end using cross-entropy loss and an Adam optimizer.

UMAP plots and semantic clusters. For the UMAP plots in Figure 1 and Figure 3, the categories are created by using Gemini 2.5 Pro to split the top 4000 words into categories, with each category inspected and refined by hand. The exact word to category mappings can be found in the code provided in the supplementary material. The UMAP plots are made with 15 neighbors, a min distance of 0.01, and a cosine metric.

Linear probes. In this work, we argue that non-linear AEs can sometimes linearly encode low-rank structure of the data. To quantify how *linear* the representations learned by the ReLU-AE are, we deploy a simple linear probe. After fully training the ReLU-AE, we freeze its encoder and collect the latent activations $\mathbf{h} = \mathbf{W}\mathbf{x} \in \mathbb{R}$. A probe is a single linear layer $\mathbf{P} \in \mathbb{R}^{V\times m}$ that is trained from scratch to reconstruct the input without any non-linearity:

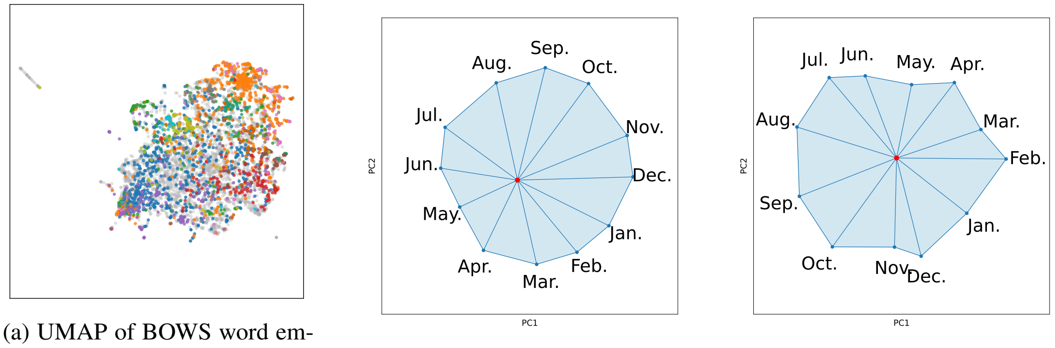
$$\hat{\mathbf{x}}_{\text{probe}} = \qquad \mathcal{L}_{\text{probe}}(\mathbf{x},\mathbf{P}) = \big\|\mathbf{x} - \hat{\mathbf{x}}_{\text{probe}}\big\|_2^2. \tag{8}$$



(a) UMAP of BOWS word embeddings

(b) PCA of unembedding weights

Figure 8: **Tow-layer transformer trained on Wikitext-103 for the multi-task token-recovery task exhibit semantic clusters and ordered circular representations for the months of the year.** This replicates the main results in Figure 4 and Figure 3 on a different dataset. Colors correspond to different semantic categories (Figure 3d).



(a) UMAP of BOWS word embeddings

(b) PCA of the data

(c) PCA of the weights

Figure 9: **Autoencoders trained on OpenWebText exhibit semantic clusters and ordered circular representations for the days of the week.** This replicates the main results in Figure 4 and Figure 3 on a different dataset. Colors correspond to different semantic categories (Figure 3d).

To measure how much of the ReLU-AE's predictive power can be captured by a linear mapping we use the Fraction of Explained Variance (FEV) of the probe relative to the ReLU-AE:

$$\mathrm{FEV} = 1 - \frac{\sum_i \left\| \hat{\mathbf{x}}_{i,\mathrm{ReLU}} - \hat{\mathbf{x}}_{i,\mathrm{probe}} \right\|_2^2}{\sum_i \left\| \hat{\mathbf{x}}_{i,\mathrm{ReLU}} - \bar{\mathbf{x}}_{\mathrm{ReLU}} \right\|_2^2}, \tag{9}$$

where i indexes data points and $\bar{\mathbf{x}}_{\mathrm{ReLU}}$ is the mean ReLU-AE reconstruction over the evaluation set. An FEV of 1 indicates that a *purely linear* map can reproduce the ReLU-AE's outputs perfectly; an FEV of 0 means the probe does no better than predicting the mean. We report the FEV on a held-out validation split.

B Toy Transformer setting

We train a two-layer transformer on WikiText-103 to predict, at each position, the set of vocabulary items that has appeared so far in the causal context.

Architecture. One encoder block with pre-norm attention and MLP:

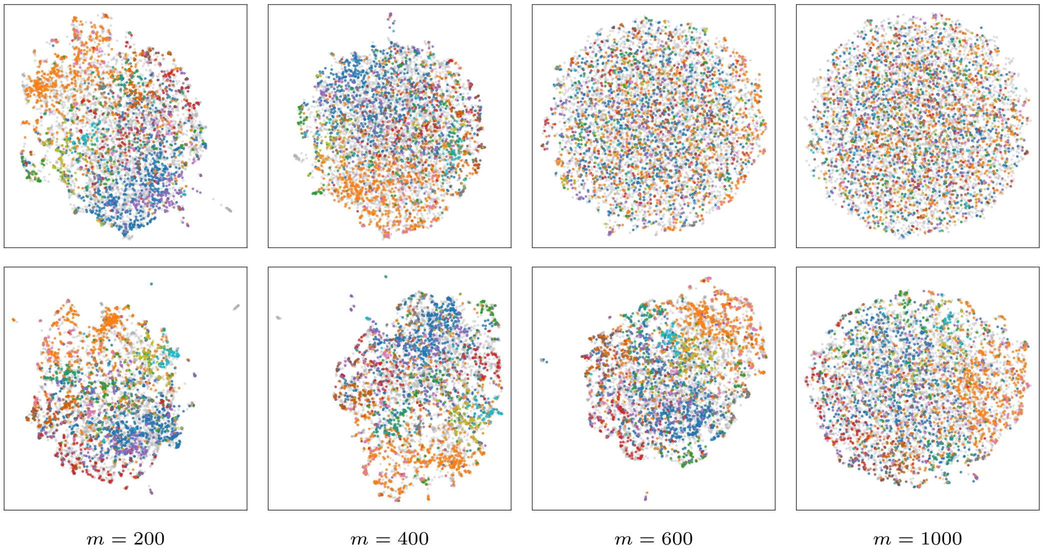


Figure 10: UMAP embeddings of features from AEs trained with context size of 20 records with wd=0 (top) and wd=4 (bottom) across different latent sizes. The plot shows that semantic structure remains for a larger fraction of context sizes when using weight decay. Colors correspond to different semantic categories (Figure 3d).

- Token embedding dimension $d_{\text{model}} = 768$, tied to the output projection.
- Multi-head self-attention with 8 heads, dropout 0, causal masking, and learned positional embeddings.
- Feed-forward network of width $4d_{\text{model}}$ (GELU activation), followed by layer normalization.

Data and tokenization. WikiText-103 is tokenized at the word level with a fixed vocabulary of $16{,}000$ tokens. The tokenizer and reserves `<pad>` and `<unk>`. Sequences are constructed with window length 512 and stride 512, padding to full length.

Targets. For each sequence position, the target is a multi-hot vector over the vocabulary indicating whether the token has appeared anywhere in the prefix (inclusive). This is computed from the sequence with padding tokens masked out.

Loss and optimization. Training uses binary cross-entropy with logits over the multi-hot targets. Optimization uses AdamW with learning rate 3×10^{-4}, weight decay 5×10^{-2}, batch size 8, and cosine annealing schedule. Gradients are clipped to norm 1.0.

In Figure 8 we see that, similarly to the BOWS setup, semantic clusters and circular structures appear in the residual stream of the transformer as a byproduct of compression.

C PCA & GEOMETRY PRESERVATION

We compute correlations in data by the *Pearson correlation coefficient*. A *Pearson correlation matrix* $\mathbb{R}$ is a *Gram* (inner-product) matrix of the standardized vectors. Due to the normalization, the Euclidean distance (chordal) distance monotonically links to the angular distances:

$$d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = 1 - \cos\left(\mathbf{x}_i^\top \mathbf{x}_j\right) = 2\left(1 - R_{ij}\right),$$

where $\|\mathbf{x}_i\| = 1$, and R_{ij} is the i^{th} and j^{th} entry of R. So correlation induces Euclidean geometry (up to rotation/reflection) on the embedded points. Hence, performing PCA on R is equivalent to performing classical *multidimensional scaling* (MDS) (Cox & Cox, 2008) on chordal distances, which explicitly aims to embed vectors such that between-vector distances are preserved as well as possible. Hence PCA yields embeddings which reflect the geometry induced by the correlations in data.

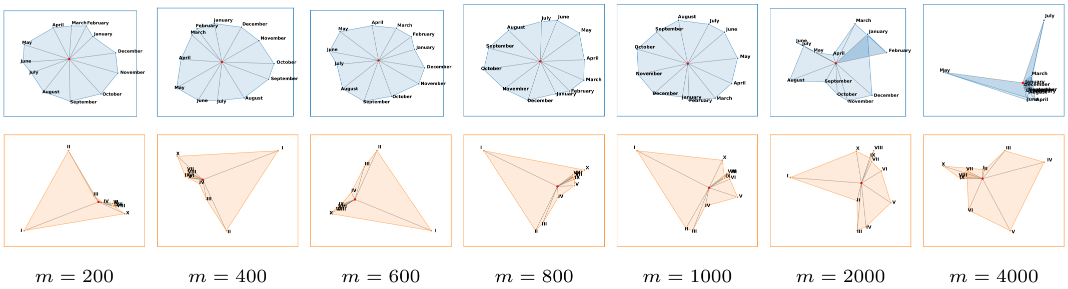


Figure 11: Reconstructions at different latent-vector sizes. Top: "Months" dataset; middle: "Roman numerals"; bottom: corresponding latent size m.

D OPENWEBTEXT REPLICATIONS

In Figure 9 we replicate the main results of the paper showing that the appearance of circular structure for the months of the year and semantic clusters is not a phenomenon limited to WikiText. These results are from an OpenWebText BOWS setup with $v = 10,000$, $c = 10$ and a stride of 10. Similarly to the WikiText case studied in the main text, we observe semantic clustering of word embeddings and that circular structures appear both when taking the PCA of the data and the trained encoder weights.

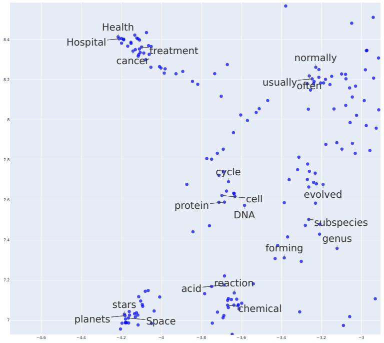


Figure 12: Zooming into the cluster for science features in the UMAP plot with a latent size of 200, we observe subsclusters within it. Medical features are in the top left while astronomy features are in the lower left and chemestry features are in the lower center.

E WEIGHT DECAY AND SUPERPOSITION

In Figure 10 we show an extended comparison of the UMAP plots presented in Figure 3, comparing models trained with weight decay (top) and without it (bottom) across different latent sizes. The results show that semantic structure is visible in UMAP plots across a wider range of latent sizes in models trained with weight decay.

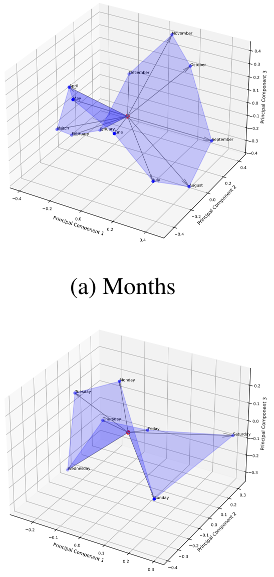
F MORE DETAILED EXAMPLE OF GROUPS OF FEATU

In the main paper we only show the feature structures for some representative latent sizes due to space constraints. In Figure 11 we show the structures studied in Figure 6 for an extended range of latent sizes.

We also show a zoomed in version of one of the UMAP plots in Figure 12. This figure highlights the rich structure of the features beyond simple clustering of high-level classes. We see that words corresponding to sciences are clustered together, but within this high level cluster, sub-groups like words about medicine (top left), astronomy (lower left), chemistry (lower center) and biology (center) are also grouped in smaller clusters.

F.1 SOME EXAMPLES BEYOND 2D

Beyond the 2D examples presented in the main paper, we include 2 examples showing that the days of the week and months of the year have structure beyond a 2D circle (Figure 13). This is clear in the case of the months where an ondulation in the third principal component is present beyond the 2D circular structure.



(a) Months

(b) Days

Figure 13: 3-D PCA of the embeddings for the words and the days in a WikiText BOWS setup.

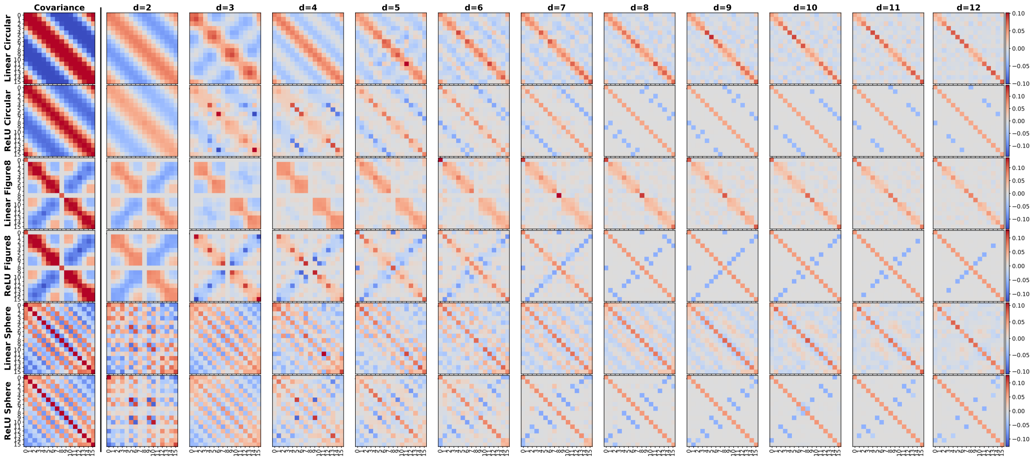


Figure 14: Extension of Figure 1 to include all values of m between 2 and 12, as well as a comparison with the weight patterns for AEs trained on data drawn from data with different correlation strucutres: figure-of-eight and spherical.

G Other correlation structures in synthetic data

In Figure 14 we show the superposition patterns for the values of m missing in Figure 1, as well as examples for autoencoders trained on data with a figure-of-eight correlation structure or a spherical structure. In all 3 cases we see that the Gramm matrix structure is similar in the linear and ReLU cases for $d = 2$ and $d = 3$ but they diverge at larger latent sizes as the ReLU-AEs start leveraging non-linear superposition which is indicated by sparse interference patterns (for example for $d = 8$).

We reuse the “synthetic months” pipeline described in Appendix A but change only the latent curve $z(\cdot)$ and the feature directions W; Steps 1 (phase selection), 4 (Bernoulli sampling), and the log-odds $\ell_k = \beta\, W_k^\top z + b$ are identical, with defaults $\beta = 5.0$, $b = -2.0$, noise $= 0.1$, seed $= 42$.

For the *Figure-8 (Lissajous)* (Lawrence, 1972), we replace Steps 2–3 by:

$$z(\theta) = \begin{bmatrix} \sin\theta \\ \sin(2\theta) \end{bmatrix}, \qquad W = \Big[(\sin\varphi_k,\ \sin(2\varphi_k)) \Big]_{k=0}^{F-1},\ \varphi_k = \tfrac{2\pi k}{F}.$$

For the *Sphere* $(\mathbb{S}^2)$, we replace Steps 2–3 by a 3D unit-sphere embedding:

$$z \sim \mathrm{Unif}(\mathbb{S}^2), \quad W = \Big[(\cos\theta_k \sin\phi_k,\ \sin\theta_k \sin\phi_k,\ \cos\phi_k) \Big]_{k=0}^{F-1},$$

where a Fibonacci lattice (Marques et al., 2013) gives approximately uniform feature directions:

$$\phi_k = \arccos\Big(1 - \tfrac{2(k+0.5)}{F}\Big), \qquad \theta_k = \pi(1 + \sqrt{5})\,(k + 0.5).$$

Full implementation details are provided in the supplementary material.

H A tail of partially reconstructed features

An interesting observation is that some features seem to appear in the correct semantic cluster while the AE is only able to capture a small fraction of their variance (e.g. $R^2 < 0.3$). In Figure 15 we show that wether we filter for features with lower or higher reconstruction scores ($R^2 < 0.3$ or $R^2 > 0.3$) they still form semantic clusters. An explanation for why features with very small R^2 scores seem to have semantically meaningful representations is that, if a model is learning principal components of the data, it might project all the features, even uncommon ones, onto these principal components. This would mean that the representations of these features is only their projection onto some principal components, even if they only explain a small fraction of their variance, explaining the observed structure in poorly represented features.

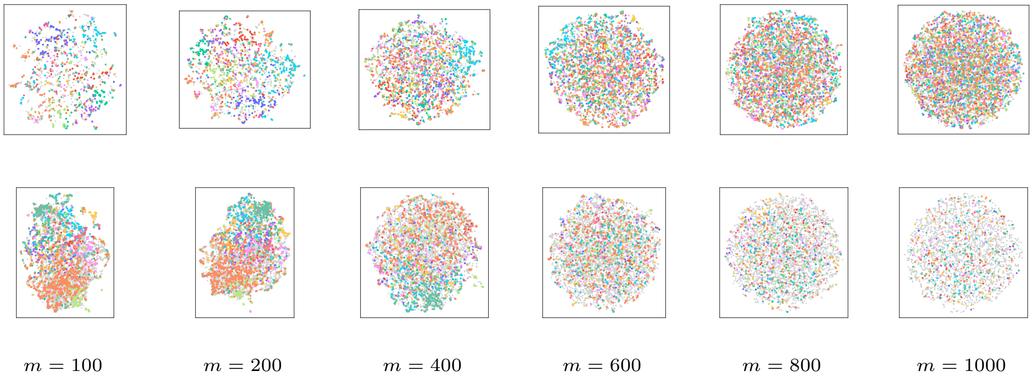


Figure 15: UMAP embeddings at different latent-vector sizes including only features with $R^2 < 0.3$ (top) and $R^2 > 0.3$ (bottom). Semantic clusters at different latent sizes are still observed in both, although this effect is combined with an increase in the number of features above the threshold in the lower one.

I Implications for the linear representation hypothesis

While the linear representation hypothesis (LRH) is one of the pillars of current mechanistic interpretability (MI) approaches. There is still no consensus on the correct formulation of this hypothesis. The LRH can be taken to mean that internal features of a model correspond to activations along one-dimensional directions in activation space (Engels et al., 2025). However, the LRH can also be formalized around the mathematical notion of linearity meaning the representation of two features is the addition of their representations and scaling a feature corresponds to scaling its representation (Elhage et al., 2021).

While some works have suggested that observed feature geometry like the ordered circles formed by the months undermine the first definition (Engels et al., 2025; Sharkey et al., 2025), our results show that these structures can emerge from the compression and reconstruction of one-dimensionally linear features. This means that these structures do not necessarily undermine either formulation of the LRH. On the other hand, our results in Section 5 do suggest that some features used by DL models can be value-coding meaning they can encode concrete trigonometric values or coordinates along linear directions which do not fulfill the constraints for mathematical linearity. For example scaling the value of a cosine-coding feature leads to a different (and potentially invalid) cosine value, rather than a stronger activation of the same cosine value.

An interesting line of research would be to explore if presence-coding features can have value-coding components. Findings like the fact that city representations in language models can be projected linearly onto a coordinates subspace (Gurnee & Tegmark, 2024), or that integers can be projected onto a helix subspace (Li et al., 2025) could be understood through this lens. In this view, city representations could have a coordinate-coding component and integers could have a size-coding component as well as sine and cosine coding components which combine to make a helix structure.

Table 2: Validation MSE and linear VE ratio for ReLU sparse auto-encoders (mean ± SD over three seeds).

Latent	MSE		Linear VE ratio	
size	mean	± SD	mean	± SD
50	0.021	$\pm 4.500{\times}10^{-6}$	0.765	± 0.003
100	0.019	$\pm 1.920{\times}10^{-6}$	0.756	± 0.002
200	0.018	$\pm 7.080{\times}10^{-6}$	0.738	± 0.000
400	0.015	$\pm 7.000{\times}10^{-6}$	0.693	± 0.000
600	0.013	$\pm 8.780{\times}10^{-6}$	0.659	± 0.001
800	0.011	$\pm 1.510{\times}10^{-6}$	0.632	± 0.000
1000	0.010	$\pm 4.320{\times}10^{-6}$	0.616	± 0.000
2000	0.004	$\pm 2.400{\times}10^{-5}$	0.624	± 0.000
4000	0.001	$\pm 4.700{\times}10^{-6}$	0.756	± 0.000
6000	0.000	$\pm 2.340{\times}10^{-7}$	0.867	± 0.000
10000	0.000	$\pm 9.960{\times}10^{-8}$	0.997	± 0.001

Overall, our findings show that rich feature geometry can be explained away by linear superposition recovering the structure inherent in the data, without appealing to non-linearly encoded information with a functional role in calculation. However, we believe the existence of value-coding features could be in conflict or an exception to features being mathematically linear.

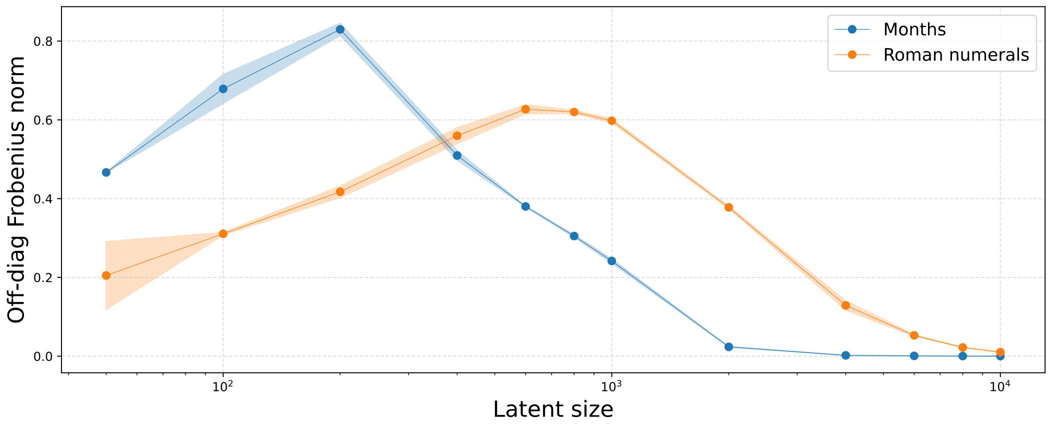


Figure 16: A reproduction of Figure 6a with the filled in areas reflecting two standard deviations across 3 seeds.

J Variation across seeds

We now provide some additional results with the variance across seeds for the main numerical results in the paper. In Figure 16, we show a version of the curves in Figure 6a, with shadded areas corresponding to two standard deviations across 3 seeds. This highlights that the order in which feature geometry appears and disapears for different features is maintained across seeds. We also provide the standard deviations for the values in Figure 1 across 3 seeds in Table 2.

K Other dimensionality reduction methods

To verify that the semantic clusters are not dependent on the choice of dimensionality reduction method, we include t-SNE (van der Maaten & Hinton, 2008) and PaCMAP (Tuncer et al., 2015) as two alternatives in fig. 17.

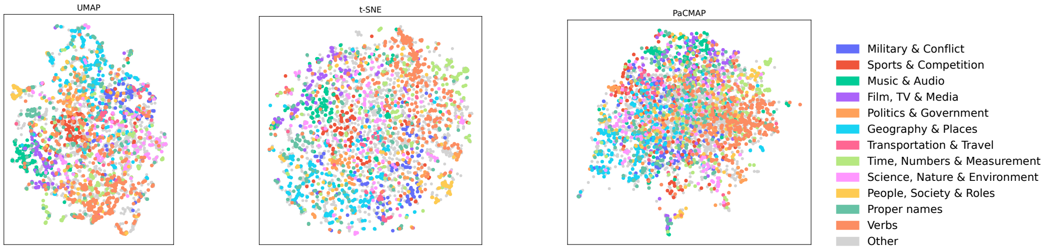


Figure 17: We show the latent representations of the top 4000 most frequent words using UMAP (left) t-SNE (middle) and PaCMAP (right) to highlight that these semantic clustering results are not dependent on the choice of dimensionality reduction technique.