Geometry of Concepts in Next-token Prediction: Neural-Collapse Meets Semantics

Yize Zhao[1], Christos Thrampoulidis[1]
[1]The University of British Columbia
zhaoyize@ece.ubc.ca, cthrampo@ece.ubc.ca

Modern language models, trained through the conceptually simple next-token prediction (NTP) objective, demonstrate a remarkable ability to capture meaning despite being trained only on explicit (context, next-word) pairs. This raises a fundamental question: How do these models extract and encode latent concepts—such as semantic dichotomies like true/false and male/female, or grammatical distinctions like nouns/verbs—during training? We discover that these latent concepts are inherently encoded in the singular value decomposition of a data sparsity matrix, which captures the support structure of conditional next-word probabilities. While NTP training never explicitly constructs this matrix, the emergent word and context embeddings naturally factor it, thereby capturing linguistic structure. Our results reveal a new form of neural-collapse geometry of latent concepts in NTP that goes beyond traditional geometry of embeddings studied previously in balanced one-hot classification settings. Furthermore, while sharing conceptual similarities with classical distributional semantics, our results reveals how neural models can acquire semantic concepts during training without explicitly constructing co-occurrence matrices.

1. Introduction

The remarkable ability of language models to capture meaning raises a fundamental question: *How do these models encode information from natural language training data into representations that enable their impressive capabilities?* Consider modern causal models trained with autoregressive next-token prediction (NTP), where the objective is conceptually simple: for each context (sequence of preceding tokens) in a text corpus, minimize the cross-entropy loss between predicted and actual next tokens. At the end of training, the model learns d-dimensional vector representations for each token (called word embeddings) and each context (called context embeddings). This naturally leads to the question: *How is the geometry of these representations of words and contexts learned by NTP training determined by the statistics of the training data?*

Recent work Zhao et al. [2024] has shown that when the model is sufficiently large and well-trained, NTP training yields word and context embeddings that correspond to matrix factorization of (a centered version of) what we call the data sparsity matrix—a binary matrix where entry (z, j) is 1 if and only if word z appears as the next token of the j-th context in the training corpus.

While this finding provides insight into how models encode explicit training signals, it opens up a deeper question about the mechanisms of latent semantic learning. Language, presented to the model as explicit (context, word) pairs, carries meaning through latent semantic information. Thus, we ask: *How do models trained with NTP extract and encode latent concepts?* For instance, how do they capture semantic concepts like true/false and male/female, or grammatical concepts like nouns/verbs? This question is particularly challenging because, unlike the explicit (context, next-token) pairs captured in the data sparsity matrix, the concepts are never directly observable by the NTP training objective.

Our key finding reveals that the encoding of latent linguistic concepts through NTP training follows a remarkably simple mechanism: the learnt concepts are inherently encoded in the singular value decomposition (SVD) of the centered data sparsity matrix. Specifically, we show that the principal

Second Conference on Parsimony and Learning (CPAL 2025).

components of the data sparsity matrix correspond to distinct grammatical and semantic concepts, with singular values quantifying their significance, and singular vectors capturing how these concepts manifest in words and contexts. Crucially, while NTP training never explicitly constructs or decomposes the data sparsity matrix, the emergent word and context embeddings naturally factor this matrix Zhao et al. [2024], thereby capturing latent linguistic structure.

Our finding makes individual contributions and establishes a link between two previously disconnected literatures:

First, our results provide new insights into the literature on distributional semantics by explaining how NTP training implements semantic encoding through an implicit SVD factorization of a data matrix. While this shares similarities with classical latent semantic analysis our message differs in two crucial ways: (1) NTP training never explicitly constructs a data membership matrix of embeddings/next-tokens or explicitly computes its SVD, and (2) the data membership matrix that NTP implicitly processes to acquire semantic information is a centered data sparsity matrix, distinct from the classical count-type matrices considered in the LSA literature.

Second, we extend the literature on neural collapse geometries beyond its traditional focus on balanced, one-hot supervised classification settings . By analyzing NTP in language modeling as an inherently imbalanced multilabel problem, we reveal the rich structure of the data sparsity matrix: its SVD factors, when examined feature-wise, expose the semantic structure present in the data.

2. Background

2.1. Setup: NTP Objective as Sparse Soft-Label Classification

We define vocabulary $\mathcal{V} = [V] := \{1, \ldots, V\}$, where $z_t \in \mathcal{V}$ represent tokens/words within sequences $\boldsymbol{z}_{1:t} = (z_1, \ldots, z_t)$. The NTP task is to predict a target token $z := z_t$ from context $\boldsymbol{x} := \boldsymbol{z}_{1:t-1}$ using training data $\mathcal{T}_n := \{(\boldsymbol{x}_i, z_i)\}_{i\in[n]}$, where $\boldsymbol{x}_i \in \mathcal{X} := \mathcal{V}^{t-1}$ and $z_i \in \mathcal{V}$ for each $i \in [n]$, and the context length $t-1$ ranges from 0 to $T-1$.

A model $f_{\boldsymbol{\theta}'} : \mathcal{X} \to \mathcal{V}$ is trained, where $f_{\boldsymbol{\theta}'}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x})$, with $\boldsymbol{W} \in \mathbb{R}^{V\times d}$ as the decoding matrix and $\boldsymbol{\theta}$ parameterizing the context to embedding map $\boldsymbol{h}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$. The model minimizes the empirical cross-entropy (CE) loss, using either MLP, LSTM, or Transformer (TF) architectures for embedding.

Sparse-Label Representation: Following Thrampoulidis [2024], we interpret the next-token prediction (NTP) objective as classifying among $m \le n$ unique contexts $\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_m$. Each context $\bar{\boldsymbol{x}}_j$ is associated with a sparse label vector $\hat{\boldsymbol{p}}_j \in \Delta^{V-1}$ in the $V-1$ dimensional probability simplex, representing the conditional distribution of next tokens. The sparsity is both a sampling artifact and inherent at the population level (not all tokens from the vocabulary are valid next-tokens of a given context in natural language data). We further let $\hat{\pi}_j$ denote the empirical probability for context $\bar{\boldsymbol{x}}_j$.

The NTP training objective can be expressed as:

$$\mathrm{CE}(\boldsymbol{\theta}') = \sum_{j\in[m]} \hat{\pi}_j \cdot \ell\left(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_j); \hat{\boldsymbol{p}}_j\right) , \tag{1}$$

where ℓ measures the deviation between the model's logits $\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_j)$ for context j and its corresponding soft-label vector $\hat{\boldsymbol{p}}_j$. Unless otherwise specified, we use the standard cross-entropy (CE) loss for NTP training:

$$\ell\left(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_j); \hat{\boldsymbol{p}}_j\right) = -\sum_{z\in\mathcal{V}} \hat{p}_{j,z} \log\left(\mathbb{S}_z(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_j))\right) .$$

Here, $\mathbb{S}_z(\cdot) : \mathbb{R}^V \to [0,1]$ denotes the z-th component of the softmax function $\mathbb{S}()$, which maps the model's V-dimensional logits to the $(V-1)$-dimensional probability simplex.

For later use, it is convenient to define the **support matrix** $\boldsymbol{S} \in \{0,1\}^{V\times m}$ of the conditional probability matrix $\boldsymbol{P} = [\hat{\boldsymbol{p}}_1, \ldots, \hat{\boldsymbol{p}}_m] \in \mathbb{R}^{V\times m}$. Formally, $\boldsymbol{S}[z,j] = 1$ if and only if $\hat{p}_{j,z} := \boldsymbol{P}[z,j] > 0$. For each

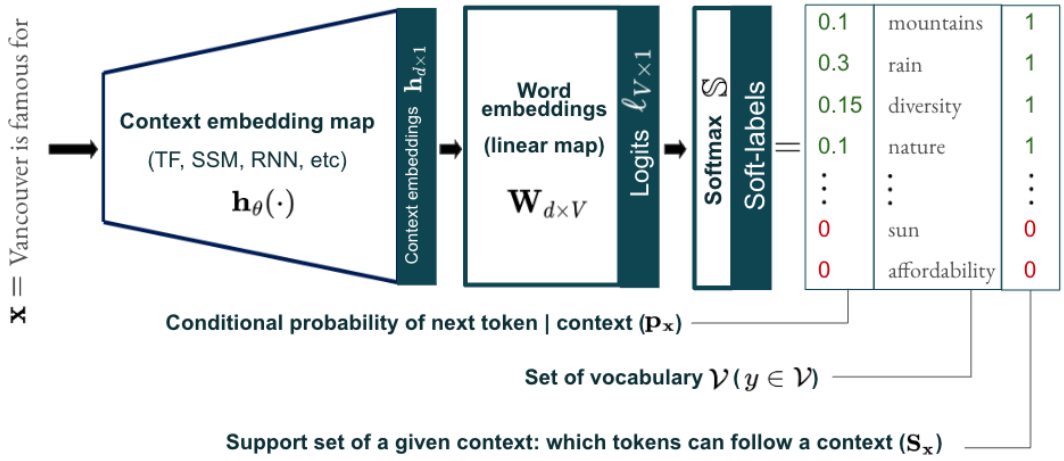

Figure 1: Notation and setup

context j, we also define its support-set $\mathcal{S}_j = \{z \in \mathcal{V} \,|\, \boldsymbol{S}[z,j] = 1\}$, which contains all tokens that appear at least once as next-tokens following context j in the training data. We refer to tokens in $\mathcal{S}_j$ as in-support tokens for context j, and all others as off-support tokens. Finally, Central to our analysis is the centered support matrix

$$\widetilde{\boldsymbol{S}} := (\mathbb{I}_V - \frac{1}{V}\mathbb{1}_V\mathbb{1}_V^\top)\boldsymbol{S} \tag{2}$$

with entries

$$\widetilde{\boldsymbol{S}}[z,j] = \begin{cases} 1 - \frac{|\mathcal{S}_j|}{V} & \text{, if } z \in \mathcal{S}_j \\ -\frac{|\mathcal{S}_j|}{V} & \text{, if } z \notin \mathcal{S}_j \end{cases}.$$

For convenience, we refer to $\widetilde{\boldsymbol{S}}$ as the **data sparsity matrix**, though note that unlike the support matrix $\boldsymbol{S}$, its entries are not binary due to the centering operation.

We illustrate the notation and setup in Fig. 1.

2.2. Geometry of Words and Contexts

Following Zhao et al. [2024], we assume sufficient model expressivity, allowing to optimize context embeddings in (1) freely, instead of abiding by their architecture-specific parameterization. This leads to the following training objective

$$\min_{\boldsymbol{W},\boldsymbol{H}} \; \mathrm{CE}(\boldsymbol{W}\boldsymbol{H}) + \frac{\lambda}{2}\|\boldsymbol{W}\|^2 + \frac{\lambda}{2}\|\boldsymbol{H}\|^2. \tag{NTP-UFM}$$

which jointly optimizes the matrices of word and context embeddings $\boldsymbol{W} \in \mathbb{R}^{V\times d}$ and $\boldsymbol{H} := [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m] \in \mathbb{R}^{d\times m}$. Since the minimization is unconstrained for both variables, we follow the neural collapse literature in referring to this as the unconstrained features model (UFM) for NTP training. The resulting log-bilinear model bears similarities to those in word2vec Mikolov et al. [2013a,b] and GloVe Pennington et al. [2014a], with two key distinctions: (1) it optimizes embeddings for both words and contexts rather than just words, and (2) it serves as a mathematically tractable abstraction of training a sufficiently expressive neural architecture. In (NTP-UFM), we have also added ridge-regularization with weight $\lambda > 0$.

Recent work Zhao et al. [2024] has analyzed the geometry of solutions to (NTP-UFM) when $\lambda \to 0$. This limit (referred to in the literature as the regularization-path Rosset et al. [2003], Ji et al. [2020]) serves as a proxy for the limiting behavior of gradient descent (GD) training as the number of

iterations approaches infinity. For large embedding dimensions $d \geq V$[1], as $\lambda \to 0$ (modeling the regime where the model is trained long-enough), Zhao et al. [2024] showed the following properties regarding word embeddings $\boldsymbol{W}$, matrix embeddings $\boldsymbol{H}$, and logits $\boldsymbol{L} = \boldsymbol{W}\boldsymbol{H}$:

1. **Logits Convergence:** The logit matrix $\boldsymbol{L}$ decomposes into two orthogonal components: (1) a sparse matrix $\boldsymbol{L}^{\text{in}}$ and (2) a diverging component aligned with $\boldsymbol{L}^{\text{mm}}$. Here, $\boldsymbol{L}^{\text{mm}}$ is the solution to a nuclear-norm minimization problem that enforces two key constraints: logits of out-of-support tokens must exceed those of in-support tokens, while all in-support tokens must have equal logits. As training progresses, the second component grows unboundedly in norm, making $\boldsymbol{L}^{\text{mm}}$ the dominant component when the logit matrix $\boldsymbol{L}$ is normalized. Importantly, $\boldsymbol{L}^{\text{mm}}$ depends solely on the data support matrix $\boldsymbol{S}$.
2. **SVD factors of $\boldsymbol{L}^{\text{mm}}$** While word and context embeddings grow unboundedly in magnitude (mirroring logit behavior), their normalized versions exhibit directional convergence. Specifically, the normalized word embeddings converge to $\boldsymbol{W}^{\text{mm}} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\mathbf{R}$ and context embeddings to $\boldsymbol{H}^{\text{mm}} = \mathbf{R}^\top\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}^\top$, where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the singular value decomposition of $\boldsymbol{L}^{\text{mm}}$ and $\mathbf{R}$ is a partial orthogonal matrix.
3. **Data-sparsity matrix as proxy:** The data sparsity matrix $\widetilde{\boldsymbol{S}}$ (see Eq. (2)) is a good proxy for $\boldsymbol{L}^{\text{mm}}$. Thus, the word and context embedings's geometries are specified by the SVD of $\widetilde{\boldsymbol{S}}$.

To sum up, Zhao et al. [2024] shows that the geometry of word and context embeddings learnt by NTP training are determined by the left and right (respectively) singular factors of the data sparsity matrix $\widetilde{\boldsymbol{S}}$. In what follows, we denote the SVD decomposition of $\widetilde{\boldsymbol{S}}$ as

$$\widetilde{\boldsymbol{S}} := \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{V\times r}$, $\boldsymbol{V} \in \mathbb{R}^{m\times r}$ with $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{V}^\top\boldsymbol{V} = \mathbb{I}_r$, and the singular values $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_r)$ are ordered:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0\,.$$

3. Geometry of Concepts

3.1. Motivating questions

In this section, we use a simple numerical example to motivate our investigation into the geometry of concepts.

Consider NTP training on the (context, next-token) pairs shown in Fig. 2, where we also display the sparse conditional probability matrix $\boldsymbol{P}$. Following Section 2.1, we train the model using objective (NTP-UFM) on this dataset. The resulting word and context embeddings, visualized through a 2D projection in Fig. 2, reveal a clear geometric structure: negative words and their associated contexts cluster on the left, while positive ones group on the right. This natural separation suggests the emergence of latent concept information—specifically, a "positive" versus "negative" semantic distinction—that shapes the embedding geometry.

As a first heuristic way to quantify this structure, we compute centroid embeddings for positive and negative examples, which we interpret as representations of "positive" and "negative" concepts. These concept embeddings form antipodal regions in the embedding space, with neutral contexts (such as "the book is" and "the movie is") positioned near their boundary. This emergent structure

[1]As noted in Zhao et al. [2024], while this assumption differs from current practice in state-of-the-art LLMs, the geometry of word/context embeddings remains rich even in this setting. Importantly, this assumption is less restrictive than requiring $d > C$ in one-hot classification settings with C classes. There, due to collapse of embeddings from the same class, $d > C$ effectively requires the dimension to exceed the number of training examples. In contrast, for NTP training, the number of contexts m can be (and typically is) much larger than the embedding dimension d, allowing for rich geometric arrangements of context embeddings in the lower-dimensional space.

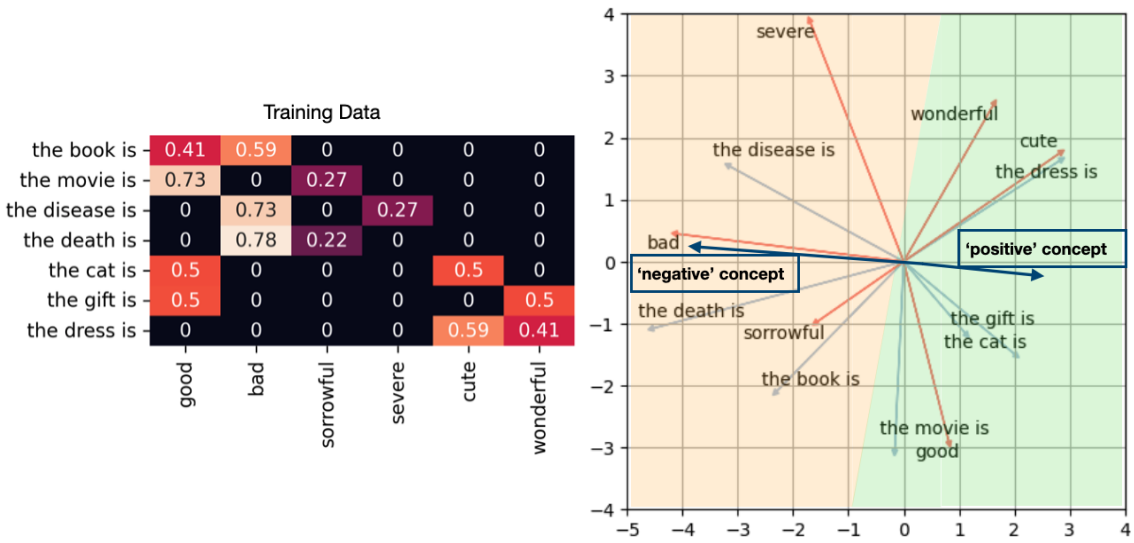

Figure 2: NTP geometry on a motivating example; see Sec. 3.1. For better illustration the synthetic training data (Left) follows the simple structure: context "the [subject] is" and next token "[attribute]." The model's embedding dimension exceeds the vocabulary size ($d > V = 10$), and the plot is generated by projecting embeddings into 2D using PCA. Word (red) and context (blue) embeddings emerge from the soft-label structure of the sparse context–next-token matrix (inset), which governs the embedding geometry. Concept representations (boxes) here are computed by averaging the related context embeddings. This paper shows how to systematically obtain the concept geometry from the SVD of a centered data-sparsity matrix.

raises two key questions: (1) How do these concept embeddings interact with and influence the geometry of word and context embeddings? (2) How does next-token prediction training lead to concept embeddings that capture semantic information?

To address these questions systematically and move beyond our initial heuristic analysis, recall from Sec. 2.2 that word and context embeddings encode the row and column space information of the training data matrix $\widetilde{\boldsymbol{S}}$. Specifically, word embeddings can be expressed as $\boldsymbol{W} = \boldsymbol{U}\sqrt{\boldsymbol{\Sigma}}\mathbf{R}$, where $\boldsymbol{U}$ comes from the SVD of the centered data-sparsity matrix $\widetilde{\boldsymbol{S}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. In other words, word embeddings are scaled and rotated versions of the left singular vectors $\boldsymbol{U}$, which encode the row-space of $\widetilde{\boldsymbol{S}}$.

The structure of $\widetilde{\boldsymbol{S}}$ provides crucial insight: its rows correspond to words, and words appearing in similar contexts—like "dog" and "cat" in sentences about animals—naturally have similar row patterns. Consequently, $\boldsymbol{U}$ may capture these semantic relationships, with its columns forming an abstract "concept basis" where a word's projection onto these axes reflects its association with specific concepts.

This observation leads to our central motivating question: Does $\boldsymbol{U}$ encode meaningful semantic information, and how does this structure emerge? Understanding $\boldsymbol{U}$ may reveal how linguistic patterns in training data shape embedding geometry and reflect latent semantic groupings.

3.2. Principal components of the data-sparsity matrix

Recall the centered sparsity matrix $\widetilde{\boldsymbol{S}}$ and its SVD in (3). Adopting terminology from Saxe et al. [2019], the columns $\boldsymbol{u}_k \in \mathbb{R}^V, \boldsymbol{v}_k \in \mathbb{R}^m, k \in [r]$ of $\boldsymbol{U}, \boldsymbol{V}$ can be thought of as **word** and **context analyzer vectors** for **concept** k. For each word $z \in \mathcal{V}$ and each word-concept $k \in [r]$, the component $\boldsymbol{u}_k[z]$ represents how *present* or *absent* is a word z in concept k. Respectively for contexts.

Each analyzer vector can be interpreted as a representation of the alignment of tokens or contexts with respect to a specific concept. Specifically, a positive value in the analyzer vector indicates that the token or context is associated with the corresponding concept. A negative value suggests that the token or context is not associated with the concept, or is oppositional to it. The absolute value represents the strength of the association. A larger positive value implies stronger alignment with the concept, while a larger negative value indicates stronger opposition.

Intuitively, a concept is a latent factor that humans—or language models—consider when processing or generating text. For example, the context "The largest city in Canada is" may embody a concept related to Canada. Words like "Toronto," "Canada," and "maple" align with this concept. This is a clear, explicit example, but not all concepts have meanings that are easily interpretable by humans. Many latent concepts captured by the model may be implicit or abstract Piantadosi et al. [2024].

To illustrate these ideas and to clearly highlight the semantic information, we constructed a simplified toy dataset using a fixed syntax as follows: "The [subject] is [attribute]", and the attribute is semantically related to the subject. The task involves predicting the last token (the attribute). This design ensures that the syntactic structure is not critical during training and, therefore, will not be extracted as a concept. Fig. 3 confirms that the concepts extracted from the SVD contain important semantic information inherent in the data.

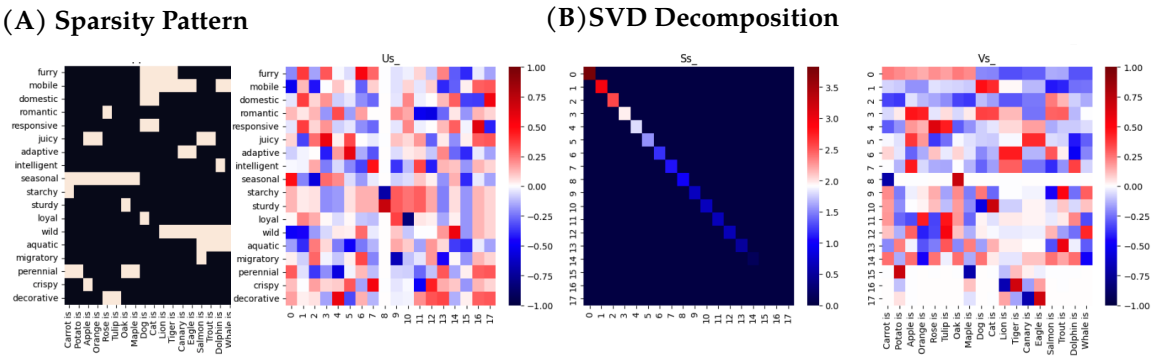

Figure 3: (**A**) The transpose of the support matrix $\boldsymbol{S}$ of a toy illustrative example, where each row represents a possible next-token and each column a context. (**B**) The SVD of the centered data-sparsity matrix $\widetilde{\boldsymbol{S}}$, illustrating how each SVD dimension may be associated with semantic meanings. For instance, in the $\mathbf{U}_s$ matrix, the first column represents the "animal-plant" dimension: tokens such as "furry" and "mobile" strongly correlate with animal traits and positively influence the first dimension, whereas "decorative" and "perennial" negatively relate, indicating plant traits. Similarly, the first row of $\mathbf{V}_s^T$ shows a significant positive alignment with "Canary is" and "Salmon is", suggesting strong animal characteristics, while negative values for "Oak is" and "Rose is" reinforce their association with plant attributes.

To find the geometry of concepts in the embedding space and relationships between concepts and the word/context embeddings, we define **word-concept representations** $\boldsymbol{u}_k^{\mathrm{d}}$ and **context-concept representations** $\boldsymbol{v}_k^{\mathrm{d}}$ for $k \in [r]$ as projections onto the spaces of word and context representations, respectively.

$$\boldsymbol{u}_k^{\mathrm{d}} = \boldsymbol{W}^\top \boldsymbol{u}_k \,, \tag{4}$$

$$\boldsymbol{v}_k^{\mathrm{d}} = \boldsymbol{H} \boldsymbol{v}_k \,. \tag{5}$$

This definition ensures that tokens or contexts more aligned with a specific concept have embeddings closer to the concept's representation, as measured by a larger dot product. Fig. 4 offers a visualization the geometry of concepts and word/context embeddings in the embedding space.

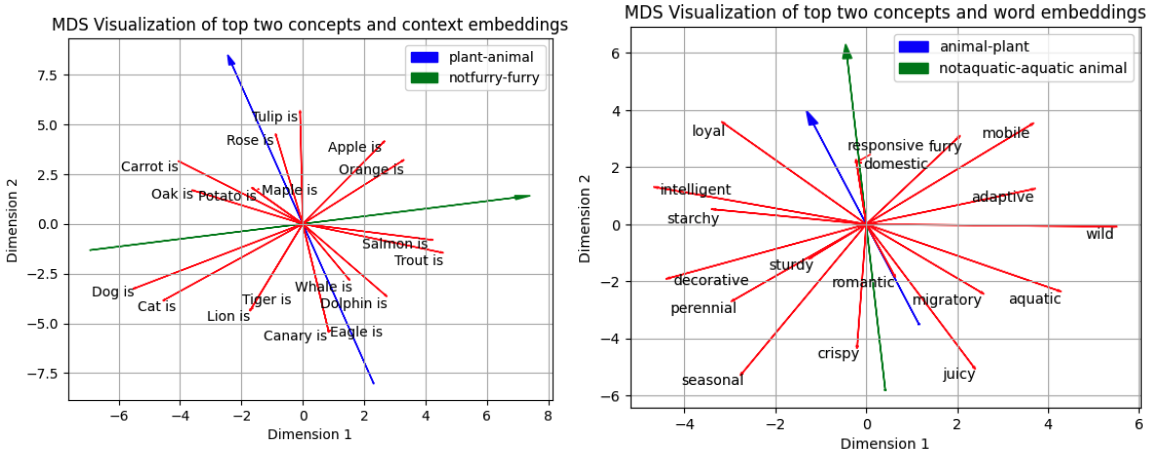

Figure 4: Visualization of (**Left**) context concepts and context embeddings, and (**Right**) word concepts and word embeddings in the embedding space. For both plots, the embeddings (red lines) represent individual words or contexts. The first and second concept axes are indicated by blue and green lines, respectively. The projection of each word/context embedding onto these axes quantifies the extent to which the word or context embodies the respective concept. For example, on the left plot, words like "Canary is" and "Salmon is" project closely along the animal end of the blue axis (animal-plant), indicating strong animal characteristics. Conversely, "Oak is" and "Rose is" align more towards the plant end, illustrating their plant-related attributes. On the right plot, the word "furry" strongly projects onto the animal side of the red axis, denoting an animal-related trait, while "decorative" projects towards the plant side, associating it with plant traits.

3.3. Rate of learning

Inspired by Saxe et al. [2019], we investigate here the rate of learning for each concept during next-token prediction training.

We train the model using objective (NTP-UFM) on a toy synthetic dataset with embedding dimension $d = 18 = V$ equal to the vocabulary size. This setup serves two purposes: the unconstrained feature model allows embeddings to optimize freely, while setting $d = V$ ensures the model can learn all concepts (represented by the rank($\widetilde{\boldsymbol{S}}$) orthonormal basis vectors from the SVD).

We track the evolution of singular values of the logit matrix during training. As shown in Figure 5, our results confirm the finding from Saxe et al. [2019]: concepts associated with larger singular values are learned faster. This pattern holds for both square loss (as in Saxe et al. [2019]) and the cross-entropy (CE) loss commonly used in NTP. While the CE loss leads to diverging singular values unlike the square loss, the relative ordering remains consistent—singular values that ultimately show maximal divergence also begin diverging earlier in training. A formal characterization of this behavior for CE loss remains an interesting direction for future work.

3.4. Hierarchical structure of the Language

By analyzing the SVD heatmaps and the learning curves of singular values generated from the toy example of Fig. 3, we observe that concepts associated with larger singular values correspond to broader categories, such as the "animal-plant" concept with the largest singular value. These broader categories are also learned faster, suggesting a progression in concept learning: general, high-level concepts are learned earlier, while finer, more specific distinctions within these categories are learned later.

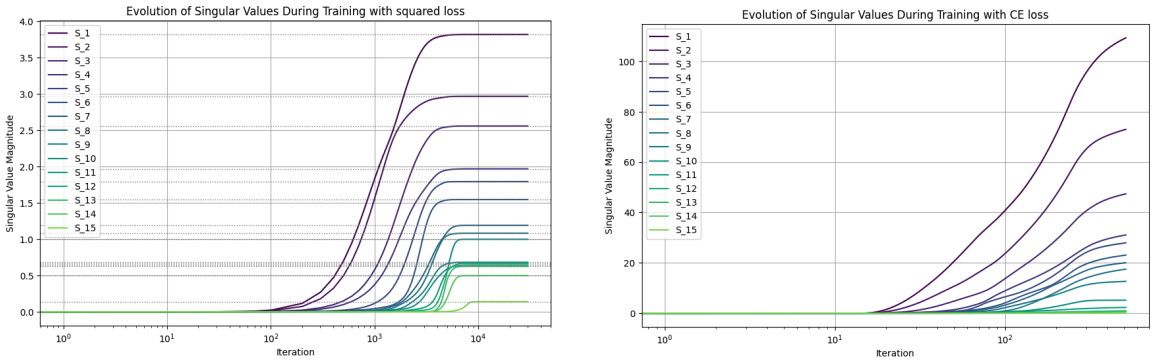

Figure 5: Evolution of singular values of the logit matrix during training. Both plots show that dominant concepts (corresponding to larger singular values) are learned first. (**Left**) With squared loss, singular values converge to those of the optimal solution, demonstrating learning saturation. (**Right**) With CE loss, singular values grow unboundedly while maintaining their relative ordering, reflecting the continuous growth of embedding norms characteristic of CE training.

To represent this progression and the relationships among concepts, we propose a tree structure. In this structure, shallower branches represent broader, more significant categories, while deeper branches represent more specific, less significant categories. Semantically similar tokens or concepts are expected to share deeper common ancestors, while dissimilar ones share only shallower ancestors.

We model this hierarchical structure using Agglomerative Clustering Müllner [2011] applied to the rows and columns of the centered sparsity matrix. Agglomerative Clustering is a bottom-up approach that starts with individual points and iteratively merges them into clusters to form a hierarchy. Here, rows of the centered data-sparsity matrix represent feature vectors for words, while columns represent feature vectors for contexts. This approach aligns with the intuition that tokens followed by more similar contexts are more semantically related, and vice versa for contexts.

The hierarchical structure constructed in this way reflects the progressive differentiation observed during learning. Shallower branches of the tree represent concepts associated with larger singular values that are learned earlier, while deeper branches capture finer distinctions learned later.

The dendrogram in Fig. 6 illustrates this hierarchy, revealing distinct clusters corresponding to broad semantic categories initially learned, with finer distinctions emerging at deeper branch levels.

3.5. Word (Context) analogy

Our formalization provides insight into the phenomenon of word analogy—linguistic comparisons of the form "A is to B as C is to D" where semantic relationships between word pairs are preserved. A classic example is "man is to woman as king is to queen," where the semantic difference captures the concept of "gender." Prior works have empirically reported that word embeddings often reflect such relationships through vector arithmetic: $\boldsymbol{w}_{\text{man}} - \boldsymbol{w}_{\text{woman}} = \boldsymbol{w}_{\text{king}} - \boldsymbol{w}_{\text{queen}}$ Mikolov et al. [2013b], Levy and Goldberg [2014], Arora et al. [2016].

We explain this phenomenon through our semantic framework. Our key insight is that semantic meaning in language manifests through individual or combined latent concepts. As demonstrated in Sec. 3.2 and Fig. 4, these concepts correspond to specific directions in the embedding space. Consequently, a change in semantics should also correspond to a linear shift in the embedding space. Therefore, if two word pairs share the same semantic change, the corresponding shifts in the embedding space should also be identical.

This intuition can be formalized using contextualized word embeddings, where word meaning depends on context. Let $\Delta \boldsymbol{s}_{z_1 - z_2} = \boldsymbol{s}_{z_1} - \boldsymbol{s}_{z_2}$ denote the semantic change between words (where $\boldsymbol{s}_z$ is the z-th row of $\boldsymbol{S}$), and $\Delta \boldsymbol{s}^{j_1 - j_2} = \boldsymbol{s}^{j_1} - \boldsymbol{s}^{j_2}$ denote the change between contexts (where $\boldsymbol{s}^j$ is

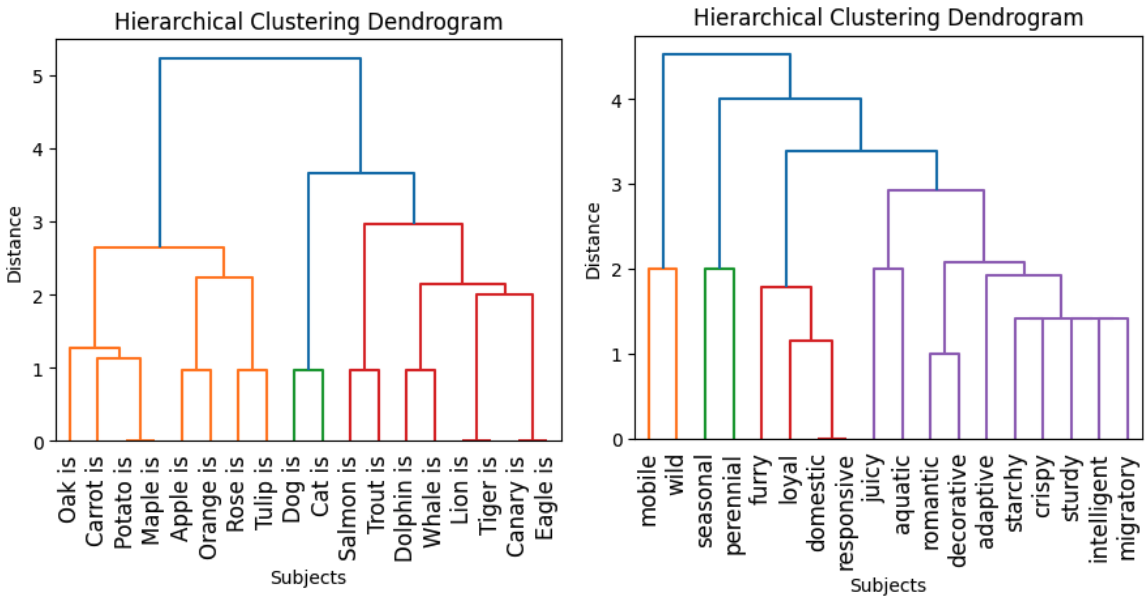

Figure 6: (**Left**) Dendrogram of subjects depicting the initial categorization phases in the semantic learning process, where fundamental distinctions between entities like animals and plants are learned first. (**Right**) Dendrogram of properties demonstrates subsequent learning stages, where specific attributes such as 'furry' or 'perennial' are distinguished, reflecting the layered complexity of semantic understanding in hierarchical learning models.

the j-th column). Our analysis demonstrates that if the semantic change between two word pairs is equivalent, then their embeddings also exhibit equivalent linear transformations. Specifically,

$$\Delta \boldsymbol{s}_{z_1 - z_2} = \Delta \boldsymbol{s}_{z_1' - z_2'} \implies \boldsymbol{w}_{z_1} - \boldsymbol{w}_{z_2} = \boldsymbol{w}_{z_1'} - \boldsymbol{w}_{z_2'}.$$

A detailed mathematical argument supporting this conclusion is provided in Appendix B. This analogy also extends to contexts.

4. Experiment

4.1. Setup

To confirm that semantic and syntactic information is encoded in the concepts extracted from the data-sparsity matrix, we conducted experiments using the TinyStories dataset Eldan and Li [2023]. We employed a limited-vocabulary word-level tokenizer, retaining only the top 500 most frequent word-level tokens, while categorizing all other tokens as "unknown." From the dataset, we extracted the 500 most common sequences of lengths between 2 and 6 tokens and calculated their empirical distributions. Using this data, we applied Singular Value Decomposition to the data-sparsity matrix to analyze its underlying structure.

4.2. Visualization and Interpretation

Building on the single-dimensional concept visualization used in the simpler toy examples in Sec. 3, we extend the analysis to a multidimensional approach to accommodate the larger scale of the dataset and the increased number of tokens, contexts, and concepts. While single-dimensional analysis is limited to broad categorizations (e.g., positive vs. negative), this multidimensional approach allows us to examine combinations of multiple semantic dimensions, providing a more detailed and nuanced understanding of the embedding geometry.

Each row of $\boldsymbol{U}$ represents a word's analyzer vector, and each column corresponds to a latent concept captured by $\widetilde{\boldsymbol{S}}$. A multidimensional approach examines how words align across multiple dimensions simultaneously. For example, instead of categorizing words into two broad groups based on a single dimension (e.g., positive or negative), we analyze combinations of k dimensions. We denote such combinations as $C = [c_1, c_2, \ldots, c_k]$, where each $c_i \in \{\text{Pos}, \text{Neg}\}$ specifies the sign of the corresponding component of the analyzer vector. For a word z, we determine its membership in a specific group defined by C as:

$$\text{Membership}(z; C) = \begin{cases} 1, & \text{if } \text{sign}(\boldsymbol{u}_{c_i}[z]) = c_i \; \forall c_i \in C, \\ 0, & \text{otherwise.} \end{cases}$$

Here, $\boldsymbol{u}_{c_i}[z]$ represents the c_i-th component of the analyzer vector for word z. This classification allows us to group words based on their alignment with multiple semantic dimensions, reflecting the multidimensional geometry of the embedding space.

4.3. Results

We find specific combinations of concept dimensions reveal strong semantic or syntactic information. While complete results for combinations of the first 3, 4, and 5 dimensions are deferred to the Appendix, we highlight a few examples below:

- **[Neg, Pos, Neg, Neg, Neg]**: This combination in the first five dimensions encodes past-tense verbs (Fig. 7 (a)).
- **[Neg, Neg, Pos, Neg, Pos]**: This combination encodes present-tense verbs (Fig. 7 (b)).
- **[Pos, Pos, Neg, Pos, Pos, Pos]**: This combination in the first six dimensions encodes prepositions (Fig. 7 (c)).
- **[Pos, Pos, Pos, Pos, Neg, Neg]**: This combination encodes proper names in the dataset (Fig. 7 (d)).

These results demonstrate that combinations of concept dimensions effectively capture complex semantic and syntactic structures in the data. The alignment of semantic groupings with multidimensional embeddings confirms the relationship between linguistic patterns and the geometry of word and context embeddings learned by NTP training.

Figure 7: Word clouds illustrating semantic information encoded by specific combinations of concept dimensions. (a) Past-tense verbs, (b) Present-tense verbs, (c) Prepositions, and (d) Proper names. Larger word sizes indicate words that are more representative of their category, emphasizing their prominence within that specific semantic configuration. Larger word sizes indicate words that are more representative of their category, emphasizing their prominence within that specific semantic configuration.

5. Discussion

5.1. $d < V$

Our analysis thus far has assumed that the embedding dimension d is at least as large as the rank of the data-sparsity matrix, which is guaranteed when $d \geq V$. Under this condition, we have shown

that word and context embeddings form a geometric structure capable of representing all concepts encoded in the data-sparsity matrix's singular factors. Moreover, as demonstrated in Zhao et al. [2024], gradient descent training naturally aligns embeddings with this structure.

However, modern language models typically employ embedding dimensions d smaller than the vocabulary size V. This raises a fundamental question: What subset of concepts can such models effectively capture? We hypothesize that during NTP training, these models learn to represent the d most significant concepts, corresponding to the largest singular values of the data-sparsity matrix.

As a preliminary investigation of this hypothesis, we trained NTP-UFM on a small synthetic dataset. Fig. 5 shows the evolution of the learned logit matrix's singular values during training, revealing convergence to the d largest singular values of the data-sparsity matrix. For this experiment, we used square loss rather than cross-entropy, as it exhibits better-behaved dynamics where singular values remain bounded during training, making their evolution more tractable to track Saxe et al. [2019]. While these initial results support our hypothesis, a deeper investigation of embedding behavior in the more practically relevant setting of $d < V$ remains an important direction for future work.

5.2. Role of Autoregression

We have shown that concepts emerge during NTP training as principal directions of the data-sparsity matrix $\widetilde{\boldsymbol{S}}$, a centered version of the support matrix $\boldsymbol{S}$. Each column j of $\boldsymbol{S}$ corresponds to a distinct context and can be viewed as a binary (multi-)label vector $\boldsymbol{s}_j$, where entries of 1 indicate tokens that appear as next-tokens for that context in the training data. The concepts learned during training are thus determined by the principal directions of this label matrix, with their importance captured by the corresponding singular values.

This analysis reveals that concepts—including semantic relationships—are implicitly encoded in the supervised component of the NTP task through interactions between context labels. At first glance, this might seem limiting: contexts can intuitively relate to each other not only through shared next-tokens (their labels), but also through the intrinsic structure of the contexts themselves, such as overlapping constituent tokens.

We argue that this second form of interaction is naturally captured through the autoregressive nature of training. In autoregressive NTP, the model processes labels for progressively longer contexts. While a context $(z_1, \ldots, z_t)$ contributes to concept formation through its distribution over next-tokens z_{t+1}, autoregressive parsing ensures that its components z_t, z_{t-1}, etc. also shape concepts through the labels of shorter subsequences $(z_1, \ldots, z_{t-1})$, $(z_1, \ldots, z_{t-2})$, and so on—each representing distinct columns in $\boldsymbol{S}$.

In other words, the rich contextual information inherent in autoregressive training manifests in concept formation through multiple, diversely interacting columns in the support matrix. The practice of forming (context, next-token) pairs through overlapping context windows naturally produces fine-grained label information, creating rich sparsity patterns in $\boldsymbol{S}$ (and consequently in $\widetilde{\boldsymbol{S}}$). These patterns yield nontrivial concepts with varying levels of significance, as reflected in their contributions to word and context embeddings.

5.3. Connection to neural collapse geometries in one-hot classification

Following our previous discussion, we formally define concepts as principal components of the data-sparsity matrix—a centered version of the support matrix $\boldsymbol{S}$ that serves as the label matrix in NTP. The richness of concepts and their varying significance emerges from the interplay of labels across different contexts.

Consider the extreme case with minimal label richness: each context is followed by exactly one next-token, with contexts distributed equally across the vocabulary. Here, $\boldsymbol{S}$ can be rearranged as $\mathbb{I}_V \otimes \mathbb{1}_{m/V}^\top$, yielding trivial concepts: all singular directions contribute equally to each word's meaning and carry equal importance, reflecting the balanced label distribution in both $\boldsymbol{S}$ and $\widetilde{\boldsymbol{S}}$.

This setting parallels standard balanced one-hot classification (with V classes and equal examples per class), as encountered in image classification. The neural collapse literature has extensively analyzed this setting, showing that last-layer embeddings (context embeddings) and weights (word embeddings) form highly symmetric aligned structures in d-dimensional space Papyan et al. [2020] (see *Related Works*). The symmetry of these geometric structures reflects the symmetric nature of $\boldsymbol{S}$, where labels induce no interesting conceptual structure.

However, such balanced one-hot settings never occur in natural language NTP Zhao et al. [2024]. Instead, as discussed in Sec. 5.2, autoregressive NTP produces rich label formations that yield diverse, interpretable concepts. In these richer settings, $\boldsymbol{S}$ induces a conceptual geometry that complements the embedding geometry studied in the neural collapse literature, thus extending its scope.

To make this connection explicit, we demonstrate that interpretable concepts emerge even in one-hot classification—the traditional focus of neural collapse literature—given minimal deviation from perfect balance. Consider the simplest imbalanced setting: STEP imbalances with ratio R, where $V/2$ majority classes each have $R > 1$ times more samples than the remaining minority classes (assuming one example per minority class for simplicity). In this setting, each column of $\widetilde{\boldsymbol{S}}$ has $V - 1$ entries equal to $-1/V$ and one entry (corresponding to the example's class) equal to $1 - 1/V$. While this structure is consistent across all examples, majority classes contribute more columns with $1 - 1/V$ entries in their corresponding rows.

Previous work Thrampoulidis et al. [2022] has shown that in this setting, the learned logit matrix converges to $\widetilde{\boldsymbol{S}}$, with embeddings emerging from its left/right singular factors. We now demonstrate how these SVD factors induce interpretable embeddings. The singular values of $\widetilde{\boldsymbol{S}}$ exhibit a three-tier structure:

$$\sigma_1 = \ldots = \sigma_{V/2-1} = \sqrt{R} > \sigma_{V/2} = \sqrt{(R+1)/2} > \sigma_{V/2+1} = \ldots = \sigma_V = 1$$

This hierarchy reveals three distinct levels of conceptual significance. To interpret these concepts, we examine the left singular vectors matrix $\boldsymbol{U}$, which as shown in Thrampoulidis et al. [2022], takes a sparse block form:

$$\boldsymbol{U} = \begin{bmatrix} \mathbb{F} & -\sqrt{\frac{1}{V}}\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{1}{V}}\mathbf{1} & \mathbb{F} \end{bmatrix} \in \mathbb{R}^{V \times (V-1)},$$

where $\mathbb{F} \in \mathbb{R}^{V/2 \times (V/2-1)}$ is an orthonormal basis of the subspace orthogonal to $\mathbb{1}_{V/2}$.[2]

The structure of $\boldsymbol{U}$ reveals three distinct types of concepts, corresponding to the three tiers of singular values: (1) The first $V/2 - 1$ columns have non-zero entries only for majority classes, representing distinctions among majority classes. (2) The middle column (with singular value $\sqrt{(R+1)/2}$) has opposite-signed entries for majority versus minority classes, encoding the majority-minority distinction. (3) The last $V/2 - 1$ columns have non-zero entries only for minority classes, capturing distinctions among minority classes.

This structure reveals a hierarchical learning process: the network first learns to distinguish between majority classes, then learns the majority-minority dichotomy, and finally learns to differentiate between minority classes.

6. Related Works

Word embeddings and semantic analysis in neural probabilistic language models. The word2vec architecture Mikolov et al. [2013a,b] and its variants, notably GloVe Pennington et al. [2014b], represent seminal early neural probabilistic language models. These simple log-bilinear models, trained on large text corpora, revolutionized word embedding learning. As noted in Zhao et al.

[2]For concreteness, $\mathbb{F}$ can be constructed using the discrete cosine transform matrix, excluding the constant column: $\mathbb{F}[i,j] = \sqrt{\frac{4}{V}} \cdot \cos\left(\frac{\pi(2i-1)j}{V}\right)$ for $i \in [V/2]$, $j \in [V/2-1]$

[2024], NTP-UFM shares structural similarities with these early models, though in both their work and ours, it serves as a tractable abstraction rather than a practical architecture. Our approach differs by learning both context and word embeddings, following modern practice. The foundational work of Levy and Goldberg [2014] connected word2vec's geometry to matrix factorization of the pointwise mutual information (PMI) matrix—a specialized word co-occurrence matrix. Subsequent works Levy et al. [2015], Turney and Pantel [2010], Baroni and Lenci [2010] empirically demonstrated semantic interpretations of the PMI matrix's singular factors and principal components. Building on Zhao et al. [2024], which formalizes a modern version of Levy and Goldberg [2014]'s results, our investigation of concepts differs from this classical literature in two key aspects: (a) We study the NTP setting where both context and word embeddings are learned, yielding concepts that relate to both words and contexts; (b) Our data-sparsity matrix differs fundamentally from classical PMI matrices: it is a centered version of the data support matrix (independent of specific next-token probabilities) and has different structural properties—being orthogonal and non-square, unlike the word2vec setting.

Superposition and feature steering Our work was partly motivated by recent compelling literature suggesting that embeddings can be decomposed into linear combinations of a finite set of semantic concepts Bricken et al. [2023], Yun et al. [2023], Park et al. [2023]. These insights from mechanistic interpretability have led to practical applications in "feature steering"—where model behavior can be controlled by manipulating concept representations through addition or subtraction Durmus et al. [2024], Konen et al. [2024]. Our analysis complements the mechanistic interpretability approach by providing a systematic framework for understanding how concepts emerge naturally as principal components from training data statistics. Exploring deeper connections between our theoretical framework and the mechanistic interpretability literature remains an intriguing direction for future work. For completeness, we note that Park et al. [2023, 2024] also investigate geometric properties of concept directions, albeit through fundamentally different technical approaches, assumptions, and perspectives, making direct comparison of our findings infeasible.

Saxe et al.'s closed-form dynamics of two-layer linear network training. Our work draws inspiration from Saxe et al. [2013, 2019]. Conceptually, Saxe et al. [2019] uses a two-layer linear neural network as a theoretical proxy to study the emergence of semantic knowledge in human cognition, providing mathematical justifications for phenomena observed in cognitive semantics literature. A key insight from their work is that even a simple two-layer linear network with orthogonal inputs can yield rich and meaningful conclusions about semantic learning. While two-layer neural networks represent perhaps the simplest instances of non-linear learning, their training dynamics generally remain analytically intractable. However, Saxe et al. [2013] (with aspects later formalized in Gidel et al. [2019]) demonstrated that with square loss, orthogonal inputs, and sufficiently small initialization, these dynamics admit exact closed-form solutions. This mathematical characterization underlies their results on semantic information development through singular factors of the network's input-output correlation matrix. We make a novel connection to this line of work: the unconstrained features model (UFM) fits perfectly within the framework studied by Saxe et al. [2013, 2019]. Specifically, the UFM can be viewed as a linear two-layer network where the input dimension equals the number of input examples (in our case, the number of contexts m). This connection is valuable in two directions: First, the UFM—recently popularized through neural collapse literature (see below)—provides perhaps the most natural and practical setting satisfying Saxe et al.'s seemingly restrictive orthogonal input assumptions. Second, this connection allows us to leverage Saxe et al.'s earlier results in the evolving neural collapse literature. Despite these methodological similarities with Saxe et al. [2019], our work differs in motivation and interpretation. We focus specifically on NTP and how semantic and grammatical concepts emerge from natural language data, rather than general cognitive development. Additionally, we primarily focus on CE loss which is typically used in NTP training.

Neural-collapse geometries. Our results contribute to the recent literature on the neural collapse (NC) phenomenon Papyan et al. [2020]. Originally observed in one-hot classification training of DNNs, neural collapse describes two key properties of well-trained, sufficiently expressive DNNs:

(1) NC: embeddings of examples from the same class collapse to their class mean, and (2) ETF-geometry: class-mean embeddings form a simplex equiangular tight frame (ETF), being equinorm and maximally separated, with classifier weights exhibiting the same structure and aligning with their respective class-mean embeddings. This phenomenon, consistently observed across diverse datasets and architectures, has sparked extensive research interest, generating hundreds of publications. One fundamental direction, which forms the basis for many extensions, focuses on explaining NC's emergence through the unconstrained features model (UFM). This model abstracts training as joint optimization of last-layer embeddings (unconstrained by architecture) and classifier weights Mixon et al. [2022], Fang et al. [2021]. Multiple influential works have proven NC emergence by analyzing the UFM's global optima Zhu et al. [2021], Súkeník et al. [2023], Han et al. [2021], Tirer and Bruna [2022], with extensions to various loss functions beyond cross-entropy, including square loss Zhou et al. [2022a] and supervised contrastive loss Zhou et al. [2022b]. Most early works maintained the original assumptions from **?**: balanced data (equal examples per class) and embedding dimension d exceeding the number of classes C. Recent work has explored $d < C$ settings, though often requiring additional assumptions on the loss function Jiang et al. [2023], Liu et al. [2023]. More substantial progress has emerged in the $d > C$ regime with unbalanced data, where Thrampoulidis et al. [2022] provided a complete characterization for step-imbalanced data (where examples are distributed equally within minority classes and equally within majority classes). They introduced the SELI (simplex-encoding labels interpolation) geometry, showing that logits interpolate a simplex-encoding matrix—a centered version of the one-hot encoding matrix. The embeddings and classifier vectors are then determined, up to rotation and scaling, by the singular vectors of this matrix. The SELI geometry emerges as a special case of the richer geometries characterized in the NTP setting by Zhao et al. [2024]. Together with Li et al. [2023], these works stand alone in extending geometric characterization beyond one-hot encoding—to soft-label and multilabel settings respectively. Specifically, Zhao et al. [2024] analyzes the soft-label setting arising in NTP training on natural language, showing that word and context embeddings are determined by the singular factors of the data sparsity matrix. Our work deepens this understanding by revealing that these SVD factors encode conceptual meaning, thereby extending neural collapse geometry to capture not only the structure of embeddings but also the organization of latent concepts.

7. Conclusions

This work provides novel insights into how next-token prediction training inherently encodes latent linguistic concepts, unveiling a deep connection between data geometry and semantic representation. By linking the singular value decomposition of the data sparsity matrix to the learned embeddings, we demonstrate that modern language models capture semantic and grammatical structures without explicit co-occurrence analysis or predefined concept constraints. This emergent geometry transcends classical approaches like latent semantic analysis, offering a unified framework to explain the acquisition of concepts ranging from broad categories to fine-grained ones. Additionally, our findings extend the scope of neural collapse geometries to imbalanced multilabel settings, offering a richer interpretation of embeddings and their alignment with principal semantic dimensions. This theoretical framework not only illuminates the structural properties of embeddings but also provides practical implications for interpreting and steering model behavior through concept manipulation. This work aims to deepen our understanding of the mechanisms by which LLMs learn and represent semantics, suggesting pathways for more human-like understanding in artificial intelligence.

References

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. *arXiv preprint arXiv:2408.15417*, 2024.

Christos Thrampoulidis. Implicit bias of next-token prediction. *arXiv preprint arXiv:2402.18551*, 2024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013a.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013b.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014a.

Saharon Rosset, Ji Zhu, and Trevor Hastie. Margin maximizing loss functions. In *NIPS*, pages 1237–1244, 2003.

Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, pages 2109–2136. PMLR, 2020.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23): 11537–11546, 2019.

Steven T Piantadosi, Dyana CY Muller, Joshua S Rule, Karthikeya Kaushik, Mark Gorenstein, Elena R Leib, and Emily Sanford. Why concepts are (probably) vectors. *Trends in Cognitive Sciences*, 2024.

Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27, 2014.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Christos Thrampoulidis, Ganesh R Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. *arXiv preprint arXiv:2208.05512*, 2022.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014b.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the association for computational linguistics*, 3:211–225, 2015.

Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

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.

Zeyu Yun, Yubei Chen, Bruno A Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. Evaluating feature steering: A case study in mitigating social biases, 2024. URL https://anthropic.com/research/evaluating-feature-steering.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style vectors for steering generative large language model. *arXiv preprint arXiv:2402.01618*, 2024.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43), 2021.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. *Advances in Neural Information Processing Systems*, 34, 2021.

Peter Súkeník, Marco Mondelli, and Christoph Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. *arXiv preprint arXiv:2305.13165*, 2023.

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*, 2021.

Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. *arXiv preprint arXiv:2202.08087*, 2022.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. *arXiv preprint arXiv:2203.01238*, 2022a.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. *arXiv preprint arXiv:2210.02192*, 2022b.

Jiachen Jiang, Jinxin Zhou, Peng Wang, Qing Qu, Dustin Mixon, Chong You, and Zhihui Zhu. Generalized neural collapse for a large number of classes. *arXiv preprint arXiv:2310.05351*, 2023.

Weiyang Liu, Longhui Yu, Adrian Weller, and Bernhard Schölkopf. Generalizing and decoupling neural collapse via hyperspherical uniformity gap. *arXiv preprint arXiv:2303.06484*, 2023.

Pengyu Li, Xiao Li, Yutong Wang, and Qing Qu. Neural collapse in multi-label learning with pick-all-label loss. *arXiv preprint arXiv:2310.15903*, 2023.

A. From sparsity language pattern to concepts via SVD

Recall the centered sparsity patter matrix $\widetilde{\boldsymbol{S}}$ and its SVD

$$\widetilde{\boldsymbol{S}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \quad \text{where} \quad \boldsymbol{U} \in \mathbb{R}^{V\times r}, \boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}, \boldsymbol{V} \in \mathbb{R}^{m\times r} \quad \text{and} \quad \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{V}^\top\boldsymbol{V} = \mathbb{I}_r\,,$$

and the singular values $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ are ordered:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0\,.$$

Adopting terminology from Saxe et al. [2019], denote $\boldsymbol{u}_k \in \mathbb{R}^V, \boldsymbol{v}_k \in \mathbb{R}^m, k \in [r]$ the columns of $\boldsymbol{U}, \boldsymbol{V}$ which can be thought of as **word** and **context analyzer vectors** for **concept** k. For each word $z \in \mathcal{V}$ and each word-concept $k \in [r]$, the component $\boldsymbol{u}_k[z]$ represents how *present* or *absent* is a word z in context k. Respectively for contexts.

We think of column dimensions of $\boldsymbol{U}$ as semantic dimensions that capture semantic categories

Q: How do we define **word-concept** and **context-concept** representations, i.e. d-dimensional representations of word and context analyzer vectors for various concepts?

Let $\boldsymbol{W} \in \mathbb{R}^{V\times d}$ and $\boldsymbol{H} \in \mathbb{R}^{d\times m}$ be the representations of words and contexts. We then define **word-concept representations** $\boldsymbol{u}_k^{\mathrm{d}}$ **context-concept representations** and $\boldsymbol{v}_k^{\mathrm{d}}$ for $k \in [r]$ as projections onto the spaces of word and context representations, respectively. Specifically, let projection matrices

$$\mathbb{P}_{\boldsymbol{W}} = \boldsymbol{W}^\top(\boldsymbol{W}\boldsymbol{W}^\top)^{-\dagger}\boldsymbol{W} \qquad \text{and} \qquad \mathbb{P}_{\boldsymbol{H}} = \boldsymbol{H}(\boldsymbol{H}^\top\boldsymbol{H})^{-\dagger}\boldsymbol{H}^\top$$

That is,

$$\boldsymbol{u}_k^{\mathrm{d}} = \mathbb{P}_{\boldsymbol{W}}\boldsymbol{W}^\top\boldsymbol{u}_k \tag{6}$$

$$\boldsymbol{v}_k^{\mathrm{d}} = \mathbb{P}_{\boldsymbol{H}}\boldsymbol{H}\boldsymbol{v}_k\,. \tag{7}$$

Let's now simplify these by using the known SVD representation of $\boldsymbol{W}$ and $\boldsymbol{H}$. Using this representation (i.e. use $\boldsymbol{W} \leftarrow \boldsymbol{W}_\infty$, $\boldsymbol{H} \leftarrow \boldsymbol{H}_\infty$) we compute

$$\mathbb{P}_{\boldsymbol{W}} = \mathbf{R}\mathbf{R}^\top = \mathbb{P}_{\boldsymbol{H}}.$$

Thus,

$$\boldsymbol{u}_k^{\mathrm{d}} = \mathbf{R}\mathbf{R}^\top\mathbf{R}\sqrt{\boldsymbol{\Sigma}}\boldsymbol{U}^\top\boldsymbol{u}_k = \sigma_k\mathbf{R}\boldsymbol{e}_k = \boldsymbol{W}^\top\boldsymbol{u}_k \tag{8}$$

$$\boldsymbol{v}_k^{\mathrm{d}} = \sigma_k\mathbf{R}\boldsymbol{e}_k = \boldsymbol{H}\boldsymbol{v}_k = \sum_{j\in[m]} \boldsymbol{v}_k[j]\cdot\boldsymbol{h}_j \tag{9}$$

Thus, the d-dimensional representations of word and context analyzer vectors are the same. We thus refer to $\boldsymbol{u}_k^{\mathrm{d}} = \boldsymbol{v}_k^{\mathrm{d}} = \mathbf{R}\boldsymbol{e}_k$ as the **representation** of concept k. In other words, **the k-th concept representation is given by a weighted average of word or context embeddings with weights taken by the resppective context analyzer vectors.**

B. Word (Context) Analogy

B.1. Definition

A word/context analogy is a linguistic comparison between two pairs of words or concepts, where the semantic relationship between the first pair (A and B) is mirrored by the relationship between the second pair (C and D). This can be expressed as: "A is to B as C is to D."

B.2. Claim

The "change in semantics" of a pair of contexts or words is represented by a linear shift in the embedding space, supporting the widely observed phenomenon of word analogy.

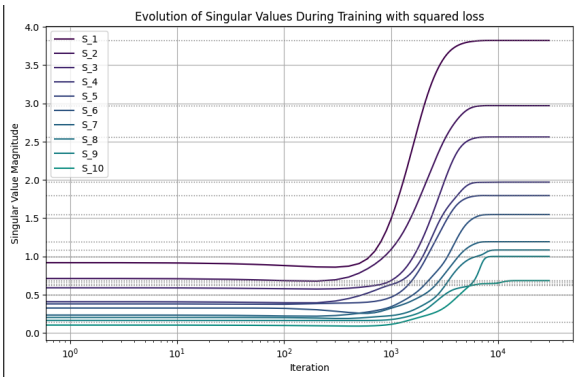

Figure 8: evolution of the learned logit matrix's singular values during training for $10 = d < V$, note that the 10 singular values are converging to the d largest singular values

B.3. Argument

Building on the idea of contextualized word embeddings—where the meaning of a word is defined by its context—we define the *change in semantics* between two words as:

$$\Delta s_{z1-z2} = s_{z1} - s_{z2}.$$

s_{z1} is the $z1$th row of $\widetilde{\boldsymbol{S}}$? Similarly, the *change in semantics* between two contexts is defined as:

$$\Delta s_{j1-j2} = s_{j1} - s_{j2}.$$

We hypothesize that if $\Delta s_{z1-z2} = \Delta s_{z1'-z2'}$, then the difference in word embeddings satisfies:

$$w_{z1} - w_{z2} = w_{z1'} - w_{z2'}.$$

To analyze this, recall the decomposition $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where:

$$\Delta s_{z1-z2} = s_{z1} - s_{z2} = \sum_{k=1}^{r} \sigma_k u_{z1,k} v_k^\top - \sigma_k u_{z2,k} v_k^\top = \sum_{k=1}^{r} \sigma_k (u_{z1,k} - u_{z2,k}) v_k^\top.$$

$\boldsymbol{v}_k$ are orthogonal, $\Delta s_{z1-z2} = \Delta s_{z1'-z2'}$ holds if and only if:

$$\sigma_k(u_{z1,k} - u_{z2,k}) = \sigma_k(u_{z1',k} - u_{z2',k}) \quad \forall k : .$$

From our theory, the embedding difference is expressed as:

$$w_{z1} - w_{z2} = (u_{z1} - u_{z2})\sqrt{\boldsymbol{\Sigma}}R.$$

If $\sigma_k(u_{z1} - u_{z2}) = \sigma_k(u_{z1'} - u_{z2'})$, then $(w_{z1} - w_{z2})\sqrt{\boldsymbol{\Sigma}} = (w_{z1'} - w_{z2'})\sqrt{\boldsymbol{\Sigma}}$. Therefore, if the analogy holds in the *support space*, it also holds in the *embedding space*.

C. Additional Experiment Results

C.1. $d > V$

C.2. More results on TinyStories

Figure 9: Analysis on combinations of 3 concept dimensions

Figure 10: Analysis on combinations of 4 concept dimensions

Figure 11: Analysis on combinations of 5 concept dimensions

Figure 12: Analysis on combinations of 6 concept dimensions