Exploring the Hidden Structures of Attention Layers in Transformer Models through the Lens of Gaussian Distributions

Martin Volker Wertich

July 01, 2024
Version: 1.0

Johannes Gutenberg University Mainz
FB08
Institute of Computer Science
Visual Computing

Bachelor thesis

Exploring the Hidden Structures of Attention Layers in Transformer Models through the Lens of Gaussian Distributions

Martin Volker Wertich

1. Reviewer Prof. Dr. Michael Wand
Institute of Computer Science - Visual Computing
Johannes Gutenberg University Mainz

2. Reviewer Prof. Dr. Ernst Althaus
Institute of Computer Science - Algorithmics
Johannes Gutenberg University Mainz

Supervisors Prof. Dr. Michael Wand and Prof. Dr. Ernst Althaus

July 01, 2024

Martin Volker Wertich

Exploring the Hidden Structures of Attention Layers in Transformer Models through the Lens of Gaussian Distributions

Bachelor thesis, July 01, 2024
Reviewers: Prof. Dr. Michael Wand and Prof. Dr. Ernst Althaus
Supervisors: Prof. Dr. Michael Wand and Prof. Dr. Ernst Althaus

Johannes Gutenberg University Mainz
Visual Computing
Institute of Computer Science
FB08
Staudingerweg 9
55128 Mainz

Abstract

Understanding the internal dynamics of Transformer models is challenging. This work aims to provide insights into why this task is so formidable. We theoretically analyze the cornerstone of the surge in Large Language Models: The attention mechanism, which adds an additional layer of complexity to an already opaque black-box model. Gladly, the embedding of human language provides us with sufficient mathematical geometrical structure, which we approximate with Gaussian distributions throughout this work.

In simple terms, two core components of utilizing Transformer models remain largely unintelligible to humans: the mathematical structure of the data and that of the learned weight matrices. We try to combine them in the context of an attention layer by intertwining Linear Algebra, Multivariate Statistics, Information Theory, and Random Matrix Theory.

A key takeaway from this work is that the concept of 'attention' in Large Language Models can be easily underestimated. This technique is not just a simple token-matching function; rather, it serves as a sophisticated combiner of marginal probability distributions influenced by their mutual dependencies, allowing the model to manage complex linear combinations through a bilinear form internally. Grasping the mathematical foundations behind attention is a significant step toward comprehending the functioning of Large Language Models.

This work is both relatively theoretical and slightly inconvenient in its structure, stemming from the complexity of this emerging field and the lack of consistent formalism. It finds its place in the intersection of Artificial Intelligence Interpretability and Natural Language Processing, which is yet to grow and manifest as a crucial pillar of Deep Learning Theory.

Contents

1 Introduction

1.1 Motivation and Problem Statement

The rise of transformer models marks a revolution in the field of Machine Learning and AI. This recent wave of success provides us with language models, which also accelerate the speed of academic research in a multifaceted way. Despite these advancements, the underlying dynamics of deep learning models remain poorly understood, prompting substantial efforts to develop abstract methods to better comprehend their capabilities in performing highly complex tasks.

Problem Statement. The domain of Transformer models is threefold complicated due to non-trivially interacting components, which I refer to as the trinity of the statistics of the data probability distribution, the behavior of the model's algorithm, and the statistics and expressiveness of the learned weight matrices. These are difficult to formulate in mathematical terms, but despite the immense availability of mathematical subject areas to cross bridges between, bringing them together coherently has not been tackled that frequently.

It is hypothesized that the data structure lies on a manifold, implying the existence of a lower-dimensional parametrization that can be approximated through dimensionality reduction techniques, such as Principal Component Analysis (PCA) utilizing Gaussian distributions. However, this approach does not account for the non-linear interdependencies that significantly influence the data manifold and are highly dependent on the dataset. Translating these interdependencies into mathematical terms that are compatible with the model's processing remains a substantial challenge that has not been fully addressed.

The expressive meaning of the weights is abstracted and compressed into incomprehensible matrices for humans. Spectral decomposition offers the advantage of estimating the impact of scaling properties on the structure of attention patterns and providing comparable orthogonal systems in their left

and right singular value bases. By randomizing the matrices' marginal elements with Gaussian distributions, following the principles of Random Matrix Theory, we can combine two intertwined random components: random data and random weight matrices. This combination enables us to adjust both entities and analyze their interdependence.

While the functioning of the attention mechanism as a token matcher is not fully understandable, examining both random components in the context of the geometry of human language raises questions about the algorithm's capabilities; it remains interesting whether we can understand attention better in terms of its capabilities if we bring all components together and whether we can exhibit patterns in attention maps by modeling the statistics of its components.

Research Gap. A notable research gap, often overlooked, is the formalization of mathematical procedures in neural networks in a manner that remains comprehensible to humans. The complexity of the problem necessitates very restrictive assumptions to derive any mathematical conclusions. Given the significant challenges posed by numerical methods, this thesis does not focus on estimating error propagation, a vast topic in itself. Instead, it justifies the approximations deemed appropriate for the sake of simplicity. Much research revolves around understanding the training dynamics, while this work looks at the inference time and tries to take a step forward to tackle the question of why a model is so powerful instead of why it becomes one.

Thesis Objective. The main objective of this thesis is to enhance the understanding of the novel attention mechanism [43] under linear and Gaussian approximation techniques utilizing Random Matrix Theory, highlighting the inherent difficulties. The attention mechanism is recognized as a particularly challenging aspect, as Transformer models are widely considered black boxes, with precise mathematical investigations of their complexity being rare. Attention maps [43] are often viewed as alignment-matching functionals [8], operating on pre-trained language embedding structure [31], which we will further explore throughout this thesis.

Explaining the formation of attention patterns is a challenging endeavor, and the current literature frequently contests their reliability [39]. Achieving a comprehensive understanding of these patterns is nearly impossible; this approach may still provide better intuition about the various factors contributing to their complex behavior and offer insight into the problem's difficulty.

Thesis Roadmap. After examining the mathematical background and the current state of the literature, this work follows four chapters in the theoretical segment, concluded by a practical series of experiments:

First, we try to scrutinize the mathematics of the data probability distribution under the lens of Gaussian models with a little touch of information theory within the constraints of linearity and realize that information theory on Gaussians becomes basically describable in terms of linear algebra and singular value theory. We also tackle how language features are mathematically captured in the joint Gaussian model.

Second, having discussed linearity, we must also address the non-linear barrier within the scope of this work. We break down the Softmax operation in a self-attention layer, which conceptually links linear combinations to asymmetric discrete probability spaces. How can we best model the nonlinearity of the Softmax operator, and how can we bridge the Euclidean vector space, whose geometry we can more easily understand, to that of the probability space? Information theory provides analogous expressions for the probability space, aiding in relating probability distributions to each other in a well-defined sense.

Third, we explore the spectral properties of the trainable weight matrices, which align with the information-theoretically driven statistical properties of Gaussians. While learned weight matrices are complex, they are descendants of random matrices and remain an active field of research due to their intricate yet statistically predictable properties. Therefore, we employ Random Matrix Theory to better understand why the learned system appears random from the outside but is complex internally. This involves tracking statistical moments for both the random Gaussians and the random matrices, considering their dependencies.

Finally, we briefly examine the mixture of probability distributions after an attention pass, which raises questions about how the model conceives linear combinations of probability distributions and how they can be dissected. That finally culminates in formulating the "The Uniform Embedding Mixture Problem" Hypothesis, connecting all three previously retrieved results to an inconspicuous but pivotal question on the path to understanding the internal model capabilities.

To apply the theoretical framework to a practical example, we inspect the development of attention maps for a Few-Shot Prompting language task in-

volving generating crossword clues for particular target words. By examining how commonly assumed principles of NLP are reflected in the attention patterns, we aim to determine if these patterns can be partly explained using our proposed mathematical framework.

2 Background

This chapter serves as a background to better understand the developed methodology in the domains of Linear Algebra, Multivariate Statistics, and Information Theory, as well as Transformer models and some Random Matrix Theory. Most information stems from the lectures that I previously attended, but I mark specific sections that stand out as overly uncommon knowledge with relevant sources.

The Linear Algebra & Multivariate Statistics sections are shaped by the respective math lectures and the "Modeling" series. The latter also laid the groundwork for the section about Information Theory. The sections about Transformer Models and the Attention mechanism follow the style of the lecture "National Language Processing" as well as the famous Vaswani et al. paper *"Attention is all you need"* [43].

2.1 Background Linear Algebra & Multivariate Statistics

Inital Data Processing & Principal Component Analysis. We start with a dataset $D = \{X_i\} \quad , \forall X_i : X_i \in \mathbb{R}^d$, represented by a matrix $X = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix} \in \mathbb{R}^{d \times n}$.

To analyze the intrinsic structure of the data, it needs to be fitted to a "linear" model, which in our case is the best Gaussian or Multivariate Normal Distribution $\mathcal{N}(\mu, \Sigma)$, with a mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, calculated via Principal Component Analysis (PCA). This method determines a set of linearly uncorrelated variables called orthogonal principal components, spanning the optimal Gaussian distribution over the data X. To archive this, the each data point X_i is shifted by the mean μ of $X_1, \dots, X_n$ so that $\forall 1 \leq i \leq n : \bar{X}_i = X_i - \mu_i$. The mean shifted data matrix is then $\bar{X} = \begin{bmatrix} \bar{X}_1 & \bar{X}_2 & \dots & \bar{X}_n \end{bmatrix}$.

The mean-shifted covariance matrix $\Sigma = \frac{1}{n-1}\bar{X}\bar{X}^T \in \mathbb{R}^{d\times d}$ is positive semi-definite and therefore diagonalizable by the spectral theorem. The eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and their associated eigenvectors are the principal components (principal axes) $v_1, v_2, \ldots, v_d$, spanning the eigenspaces $\langle v_1\rangle, \langle v_2\rangle, \ldots, \langle v_d\rangle$.

Multivariate Gaussian The shifted mean vector μ and the eigenspace of the covariance matrix $\frac{1}{n-1}\bar{X}\bar{X}^T$ make up the estimated Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, which can be interpreted as the "best Gaussian" overlaying the data - thus maximizing the variance under its shape in terms of a Maximum Likelihood Estimate. The probability density function of the Multivariate Gaussian

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

includes in the exponentiation term a quadratic form $Q(x) = (x-\mu)^T \Sigma^{-1}(x-\mu)$. The solution set to the equation $Q(x) = 0$ defines the well-known "Gaussian" ellipsoid, which is spanned by the principal components, each having the length given by the corresponding singular value of X, respectively. Along these axes, the variance of the data X is maximized. Moreover, a volume V can be associated with the ellipsoid, given by the product of the inverse singular values $V = \sigma_1^{-1}\sigma_2^{-1}\ldots\sigma_n^{-1}$. This volume measures the steepness or flatness of the Gaussian distribution and, therefore, the spread or variance of the data. Therefore, the volume of the probability distribution's ellipsoid is inversely proportional to the square root of the determinant of the covariance matrix: $\propto \frac{1}{\sqrt{|\Sigma|}} = \frac{1}{\sqrt{\prod_{i=1}^n \sigma_i}}$.

Diagonalizable Matrices. Suppose the data matrix X is diagonalizable. In that case, it can be decomposed into $X = Q\Lambda Q^T$, and the principal components can be obtained from the column space of Q, indicating that there is a one-to-one correspondence between X and the Gaussian $\mathcal{N}(\mu, \Sigma)$. If the eigenvalues in the diagonal matrix Λ are distinct, the PCA is uniquely determinable and invertible. If some eigenvalues were equal, then PCA would be uniquely determinable up to permutation for those dimensions, as their associated principal components can permute, but that should be negligible for real-valued random numbers.

Singular Value Decomposition. PCA can also be performed for arbitrary non-quadratic matrices $X \in \mathbb{R}^{n\times d}$. There exists a Singular Value Decomposition (SVD) $X = UDV^T$, with $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{d\times d}$ containing the left and right singular vectors in the column space, respectively, and $D \in \mathbb{R}^{d\times d}$ containing all

singular values in the diagonal, which are always non-negative. The squared singular values correspond in the diagonalizable case to the absolute of the eigenvalues, but the orientation is abstracted by the singular vector matrices. Therefore, the squared singular values express the absolute scaling power of the matrix. The eigenvalues of the covariance matrix Σ can be obtained by squaring the singular values. The principal components or main axes are also obtainable by the columns of V, which are the eigenvectors of the unscaled covariance matrix $\Sigma = XX^T$, whose normalized variant $\frac{1}{n-1}\hat{X}\hat{X}^T$ is the covariance matrix of the fitted Gaussian. Analogously, the column space of U represents the eigenvectors of the Gram matrix X^TX.

The PCA remains unique for distinct singular values of Σ, but it is no longer invertible, and the information contained in U is lost - the eigenspaces of the Gram Matrix $X^TX \in \mathbb{R}^{n\times n}$. This part becomes more pronounced with an increasing dataset size n. Fortunately, The correlation information, squeezed into the eigenspace of the unnormalized covariance matrix XX^T, is approximately preserved in a parametrized form.

Gaussians under Linear Maps. Lastly, if we apply a linear transformation $A \in \mathbb{R}^{n\times m}$ to each data point X_i in X: $AX = \begin{bmatrix} AX_1 & AX_2 & \dots & AX_n \end{bmatrix}$, we can observe the change of base in the Gaussian distribution from the PCA either. This results in a new Gaussian distribution $\mathcal{N}(\mu', \Sigma')$ with a new mean vector $\mu' = A\mu \in \mathbb{R}^m$ and a new covariance matrix $\Sigma' = A\Sigma A^T \in \mathbb{R}^{m\times m}$, which is the same as if applied PCA to $AX \in \mathbb{R}^m$.

2.2 Background Information Theory

After specifying the linear algebra, we delve into Information Theory. We begin with the general case for continuous and discrete variables and later progress to information theory applied to Gaussian distributions, which form the cornerstone of the work.

For simplicity, we consider the theory just for the discrete case, as the continuous case - commonplace in probability theory - is simply a transition from the summation $\sum \star$ to the integral $\int \star$, with the formula content remaining the same.

Information Theory on discrete random variables. The entropy, denoted as $H(X)$, measures the uncertainty inherent in a random variable X, defined

as $H(X) = -\sum_{x \in X} p(x) \log p(x)$, where $p(x)$ is the probability of observing state x (for Shannon-entropy, the base of $\log$ is 2, or e for Machine Learning settings).

The cross-entropy measures the dissimilarity between two probability distributions p and q and is frequently used to define the loss function for classification problems in Machine Learning. It is given by $H(p, q) = -\sum_{x \in X} p(x) \log q(x)$. The Kullback-Leibler divergence (KL divergence) is the difference between the cross-entropy and the entropy $D_{KL}(p||q) = H(p, q) - H(p)$.

Mutual information denoted as $I(X;Y)$, quantifies the amount of information obtained about one random variable through the other random variable and is mathematically defined as $I(X;Y) = \sum_{x \in X, y \in Y} p(x, y) \log \frac{p(x,y)}{p(x)p(y)}$, where $p(x, y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively. This measure also reflects how observing one of these variables reduces uncertainty about the other. Lastly, the joint entropy between two variables $\mathrm{H}(X, Y)$ is then given by $\mathrm{H}(X, Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{I}(X;Y)$.

Information Theory on Gaussians. Now, we are ready to establish information theory for Multivariate Gaussian distributions. The main idea behind this approach is to establish a bridge between multivariate statistics and linear algebra. The following formulas in this subsection are first given by GPT4 but evaluated against existing lecture sources like [@19], verifying that they are correct.

The entropy of a Gaussian distribution $p(x) = \mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$, and $\Sigma \in \mathbb{R}^{d \times d}$ is given by:

$$\mathrm{H}(p) = \frac{1}{2} \log \left((2\pi e)^d |\Sigma| \right) = \frac{1}{2} \log (2\pi e)^d + \frac{1}{2} \log (|\Sigma|)$$

The factor $(2\pi e)^d$ is a normalization factor, scaling exponentially in the dimension d of the Gaussian, and $|\Sigma| = \det(\Sigma)$, corresponding to the product of the singular values $\sigma_1, \sigma_2, \ldots \sigma_d$ of Σ (positive definite) and, thus, the volume of the ellipsoid spanned by the Gaussian distribution. This is reasonable since the structural complexity and the uncertainty in the distribution are simply reflected in the ellipsoid's volume from the quadratic form.

The cross-entropy between two multivariate Gaussian distributions $p(x) = \mathcal{N}(\mu_1, \Sigma_1)$ and $q(x) = \mathcal{N}(\mu_2, \Sigma_2)$, where , where $\mu_1, \mu_2 \in \mathbb{R}^d$, and $\Sigma_1, \Sigma_2 \in \mathbb{R}^{d \times d}$, is:

$$\mathrm{H}(p,q) = \frac{1}{2}\log\left((2\pi e)^d |\Sigma_2|\right) + \frac{1}{2}\mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) + \frac{1}{2}(\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1)$$

We now break down all three individual terms of this formula to make sense of this formula in terms of linear algebra. The first term is the second Gaussian's entropy $\mathrm{H}(q)$.

The third term $\frac{1}{2}(\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1)$ is the Mahalanobis distance between the means of the two distributions, weighted by the inverse of Σ_2, measuring how far apart the centers of the two distributions are, relative to the variance of q. Or, more informally, the mean distance in terms of the distance of the coordinate system defined by q.

The more complex second term $\frac{1}{2}\mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right)$ is easier to understand when considering both distributions through the shape of their ellipsoids, span by their principal axes. The i'th singular value encodes how much the i'th principal component of Σ_2 needed to be stretched to match the equivalent component in Σ_1. This term can be considered an aggregated principal component "stretch" score. We can also write the matrices $\Sigma_1 = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_d \\ | & | & & | \end{bmatrix}^T$ and $\Sigma_2 = \begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_d \\ | & | & & | \end{bmatrix}$ in terms of basis vectors and we actively add up the alignment or angle between the basis vectors a_i and b_i, multiplied by the length of b_i and the inverse length of a_i.

The KL divergence between p and q is then given by:

$$D_{KL}(p\|q) = \mathrm{H}(p,q) - \mathrm{H}(p)$$

The KL divergence can be considered a distance measure between two Gaussian distributions p and q. If $p = q : D_{KL}(p\|q) = 0$; otherwise, the KL distance effectively measures the "effort" required to translate from q to p. Its asymmetry suffers from drawbacks, so the Jensen distance as square root of the symmetric Jensen divergence $\mathrm{JSD}(p||q) = \frac{1}{2}(D_{KL}(p\|m) + D_{KL}(q\|m))$, with $m = \frac{p+q}{2}$ tackles that issue and the the Jensen distance also defines a metric.

Lastly, the Mutual Information between the two jointly Gaussian-distributed random variables $X \sim p$ and $Y \sim q$, given by $X = \begin{bmatrix} X_1 & \dots & X_n \end{bmatrix}$ and $Y = \begin{bmatrix} Y_1 & \dots & Y_n \end{bmatrix}$ can be calculated as:

$$\mathrm{I}(X;Y) = \frac{1}{2} \log \frac{|\Sigma_X||\Sigma_Y|}{|\Sigma|}, \quad \Sigma = \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix}$$

$$\Sigma_X = \frac{1}{n-1} X^T X, \quad \Sigma_Y = \frac{1}{n-1} Y^T X \quad \Sigma_{XY} = \frac{1}{n-1} X^T Y, \quad \Sigma_{YX} = \Sigma_{XY}^T$$

In this formula, Σ_X and Σ_Y describe the marginal covariance matrices, and Σ_{XY} and Σ_{YX} are the cross-covariance covariance matrices. To avoid confusion, X is a multivariate variable that is still marginal in the cumulative probability system, encompassing just X and Y.

If X and Y are statistically independent, $I(X;Y)$ becomes zero, rendering this parameter useless. However, if statistically dependent, the Mutual Information can be a more powerful tool than just comparing the Gaussian matrices individually because the cross-covariance matrices Σ_{XY} and Σ_{XY} cover more information than just relying solely on the marginal distributions Σ_X and Σ_Y.

The core argument of this section was to emphasize how information theory essentially becomes a Linear Algebra problem under Gaussian approximation assumptions.

2.3 Background Transformer Models

Tokenized Sequences. A Transformer [43] is an encoder-decoder model with $Enc_w(x)$ and $Dec_w(x)$ that processes a sequence $X \in [1, \mathrm{Voc}]^n$, whereas n is the sequence length and the discrete interval $[1, \mathrm{Voc}]$ is a domain, assigning each language token a unique number. In Language Transformer models, we mainly refer to Voc as the token alphabet size, making X a tokenized sequence. The tokenizer often decomposes longer words into distinct components, which are then treated individually. The next component is embeddings of size d, which deterministically convert each integer number into a corresponding real-valued vector of dimension d. However, the transformer models often include and learn that transformation in their architecture.

Embeddings. These embeddings of the tokenized sequences are pre-trained embeddings like GloVe [31] (although they are further improved during training). They provide a language geometrical structure to the language, reflecting the semantic dependencies in human language. We classify these geometric characteristics in two ways, whereas $emb(w)$ is the embedding of word $w \in \mathbb{R}^d$:

Multiplicative and Additive Compounds of Embeddings. The "multiplicative" component causes words with semantic similarity to have a high cosine similarity (e.g., $emb(\text{warm})$ and $emb(\text{sunny})$) and vice versa. Second, the "additive" component allows structurally analogous words to follow additive constraints. For example, since "Paris" relates to "France" as "Rome" relates to "Italy," we can observe: $emb(Paris) - emb(France) + emb(Italy) \approx emb(Rome)$. This also holds true for other types of language patterns, albeit with varying precision.

Although embedding specifics differ among transformer models, it is common practice that the Euclidean length $l = ||\cdot||_2$ of the embedding vectors is nearly equal, meaning that they are considered to lie on a hypersphere $\mathcal{S}_l^{d-1} = \{x \in \mathbb{R}^d | \quad ||x||_2 = l\}$ of dimension $d-1$ with radius l, but with minor distortions on the surface, when their lengths are unequal.

Positional Encodings. Moreover, sinusoidal positional encodings [7] are often added to the tokens to identify their location in the sequence and automatically increase the cosine similarity of neighboring tokens. Still, they are usually dissimilar to most other embedding vectors.

Transformer Architecture. Now, we move to the inference of a trained Transformer model, assuming the entire input sequence can be processed at once and ignoring batches; a full illustration is illustrated in graph 2.1.

A transformer is an Encoder-Decoder model $T = \{Enc(X), Dec(X)\}$, transforming a sequence X to a new sequence $Y = T(X)$. At inference time, during time step t, the Encoder takes the complete input sequence X and gets an encoded representation $Enc(X)$. The Decoder takes the last $t-1$ token of the output sequence Y and the encoded representation $Enc(X)$, predicting the next token of the output sequence $Y_i = Dec\left(Y_{[1,\ldots,i-1]}, Enc(X)\right)$. This process is repeated until a stop token is generated, so Y becomes the final output sequence.

We perform the notation for the Encoder $Enc_w(X)$ ($Dec_w(X)$ is analogous) with k transformer blocks, written as: $Enc_w(X) = T_k(T_{k-1}(\ldots(T_1(X))))$. Each

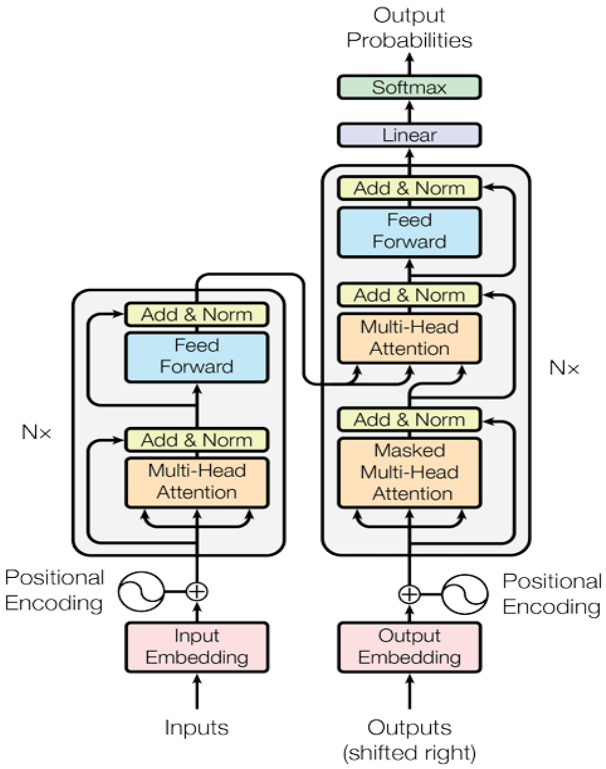


Fig. 2.1.: Original Transformer Architecture [43]

T_i represents the transformer block T_i of layer i, which can get decomposed as $T_i(x) = \text{FF}_i(\text{ATTN}_i(x))$, where FF_i denotes a feedforward neural network and ATTN_i a self-attention layer. In the self-attention layer $\text{ATTN}_i(x)$, the well-known self-attention mechanism [43, 46] takes an embedding matrix X and four attention weight matrices: the query weight matrix $W_q^{(l)}$, the key weight matrix $W_k^{(l)}$, the value weight matrix $W_v^{(l)}$ and an output weight matrix W_O are applied for an arbitrary head h in layer l.

$$\text{head}_i = \text{Self-Attention}\left(W_q^{(l)}, W_k^{(l)}, W_v^{(l)}, X\right) := \text{Attention}\left(XW_q^{(l)}, XW_k^{(l)}, XW_v^{(l)}\right)$$

head_i is applied and repeated h times in a transformer block, depending on the number h of attention heads. The results are concatenated before being fed into a following feedforward network with the RELU actuation function.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h) W^O$$

Multi-Head Transformer. Before explaining the attention mechanism in detail, it is important to note that for h attention heads in a layer l, the em-

beddings of dimension d are typically divided into portions of dimension $\frac{d}{h}$ and the final results are concatenated to restore the original dimension. Each head processes only a part of the embedding for an arbitrary token i, but the "multiplicative" and "additive" constraints from the pre-trained embeddings still apply to this fragment. Those constraints are spread across the embedding fragments in advanced embeddings, increasing representation power. Nonetheless, the output weight matrix W_O is not the scope of this study. It will be predominantly ignored for the rest of this study as we focus on Single-Head Self-Attention, which lets W_O become a simple linear layer map, making that component uninteresting to us.

Layer normalization. Additionally, it is essential to briefly mention Layer Normalization [2], which involves normalizing each column or feature dimension of the matrix of embeddings: $X = \begin{bmatrix} | & | & & | \\ \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \\ | & | & & | \end{bmatrix}^T \in \mathbb{R}^{n \times d}$ to have a mean of 0 and a variance of 1. Theoretically, the probability distribution associated with the X_i converges to the standard normal distribution according to the Central Limit Theorem as the sequence length n increases.

Skip connections. At each self-attention operation and at each feedforward linear operation, the previous embedding sequence is added to the current intermediate result with subsequent layer normalization to keep the algorithm numerically stable. This also allows the model to convey information past entire attention blocks without undergoing transformation.

Figure 2.2 illustrates the complete architecture of an exemplary transformer block, just as in LLAMA2 [40, 41] or Mistral [18].

2.4 Background Attention Mechanism

In this section, the functionality for Single-Head Self-Attention will be explained in more detail. To avoid confusion, the embeddings in the embedding matrix X are stacked in rows instead of columns, meaning $X \in \mathbb{R}^{n \times d}$ instead of $X \in \mathbb{R}^{d \times n}$, to match the dimensions in the well-known formula $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$.

The input of the attention mechanism is an embedded tokenized sequence $x \in \mathbb{R}^{n \times d}$, where n denotes the length of the sequence and d the size or dimen-

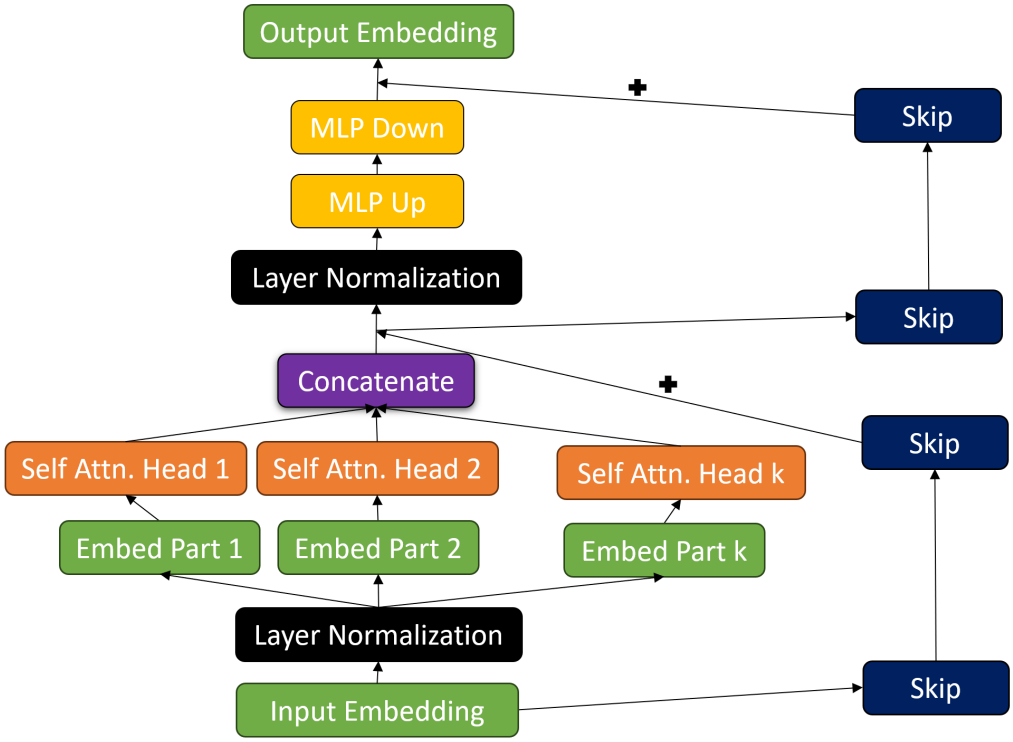


Fig. 2.2.: LLaMa/Mistral Transformer Unit Schema

sionality of the embedding, which matches the previous section using the general data matrix $X \in \mathbb{R}^{n \times d}$. The concise form of the attention mechanism [43] is:

$$\text{Self-Attention}\left(XW_q, XW_k, XW_v\right) = \text{Attention}\left(Q, K, V\right) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The primary component distinguishing Transformer models from most standard neural networks is deploying the self-attention mechanism [43], which is demonstrated for an arbitrary network layer l and attention head h. The attention mechanism utilizes three trainable weight matrices $W_q \in \mathbb{R}^{d_q \times d}, W_k \in \mathbb{R}^{d \times d_k}, W_v \in \mathbb{R}^{d \times d_v}$, whereas $d_k = d_q$, and is visualized in Figure 2.3.

Each of them gets multiplied with the embedded tokenized sequence $X \in \mathbb{R}^{n \times d}$, to embed the tokenized information into three newly embedded representations by the trainable weight matrices W_q, W_k, W_q - usually without biases.

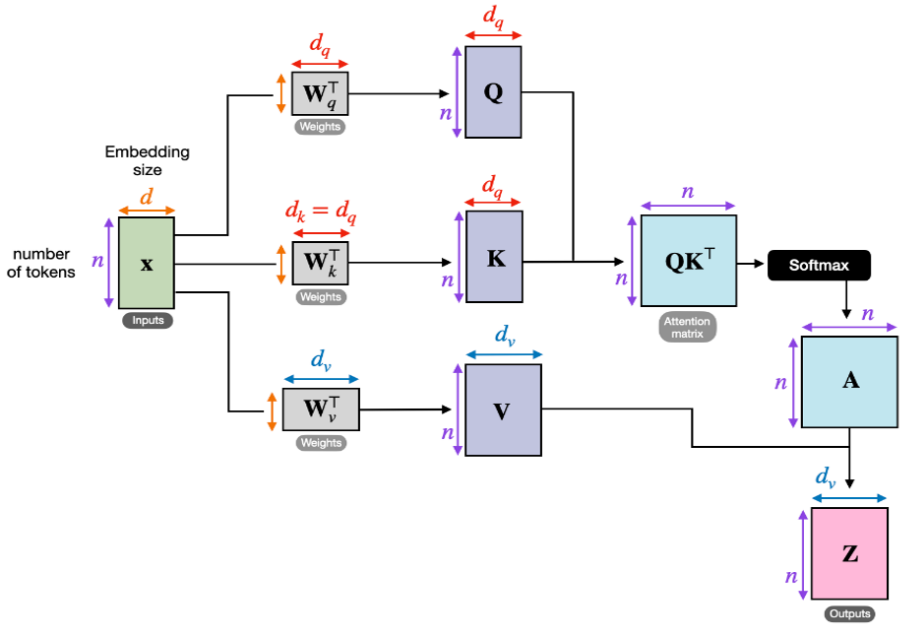


Fig. 2.3.: Architecture of Attention Layers [@34]

The resulting intermediate results are $Q = XW_q \in \mathbb{R}^{n \times d_q}$, $K = XW_k \in \mathbb{R}^{n \times d_k}$ and $V = XW_v \in \mathbb{R}^{n \times d_v}$. Therefore, we can write:

$$Q = \begin{bmatrix} | & | & & | \\ Q_1 & Q_2 & \cdots & Q_n \\ | & | & & | \end{bmatrix}^T \quad K = \begin{bmatrix} | & | & & | \\ K_1 & K_2 & \cdots & K_n \\ | & | & & | \end{bmatrix}^T \quad V = \begin{bmatrix} | & | & & | \\ V_1 & V_2 & \cdots & V_n \\ | & | & & | \end{bmatrix}^T$$

, with $Q_i \in \mathbb{R}^{d_q}$ denoting the query embedding for token i. Analogically, $K_i \in \mathbb{R}^{d_k}$ is the key embedding for token i, and $V_i \in \mathbb{R}^{d_v}$ is its value embedding.

Attention Map as Cosine Similarity Measure. The most critical aspect of the attention mechanism is the creation of the raw attention matrix $\hat{A} = \text{Softmax}\left(\frac{1}{\sqrt{d_k}} QK^T\right) \in \mathbb{R}^{n \times n}$, whereas the attention matrix element $\hat{A}_{ij}$ measures how strongly the embedding of token i attends to that of token j, referred to as raw attention score. It calculates the standard scalar product $\frac{1}{\sqrt{d_k}} \langle Q_i, K_j \rangle = \frac{1}{\sqrt{d_k}} Q_i^T K_j$ between the query embedding Q_i of token i and the key embedding K_j of token j, normalized by the dimension $\sqrt{d_k} = \sqrt{d_q}$. The normalization factor becomes explainable after the Random Matrix Theory section - it effectively corresponds to the square root of the expected length of a random vector $v \in \mathbb{R}^{d_k}$ - $v_i, i.i.d \sim \mathcal{N}(0, 1)$. To work with the attention scores statistically, the attention matrix is transformed into a row-wise stochastic

matrix A, meaning $\forall i \in 1, \ldots, n : \sum_{j=1}^{n} A_{ij} = 1$. This is achieved by applying the SoftMax operation to each row individually:

$$A = \text{SoftMax}\left(\hat{A}\right) = \begin{bmatrix} \frac{exp(\hat{A}_{11})}{\sum_{j=1}^{n} exp(\hat{A}_{1j})} & \cdots & \frac{exp(\hat{A}_{1n})}{\sum_{j=1}^{n} exp(\hat{A}_{1j})} \\ \vdots & \ddots & \vdots \\ \frac{exp(\hat{A}_{n1})}{\sum_{j=1}^{n} exp(\hat{A}_{nj})} & \cdots & \frac{exp(\hat{A}_{nn})}{\sum_{j=1}^{n} exp(\hat{A}_{nj})} \end{bmatrix}$$

The entry a_{ij} of $A = \text{SoftMax}\left(\hat{A}\right)$ is referred to as the (normalized) attention score in contrast to the raw attention scores of $\hat{A}_{ij}$.

Linear Combinations of Value Embeddings. Then the value embedding matrix V becomes multiplied by the stochastic attention matrix via $Z = AV \in \mathbb{R}^{n \times d_v}$, finally ending up in the output matrix $Z \in \mathbb{R}^{n \times d_v}$. This term is interesting as it also marks a linear approach regarding the value embeddings. If we consider Z as $\begin{bmatrix} Z_1 & \ldots & Z_n \end{bmatrix}^T$, we observe that $Z_i = \sum_{j=1}^{n} a_{ij} V_j$ simply aggregates value embeddings. This means that for the final embedding Z_i of token i, the value embeddings get linearly combined, and the percentage (stochastic matrix: $\sum_{j=1}^{n} A_{ij} = 1$) of the value embedding j being represented in the final result is the normalized attention score a_{ij} of token i to token j:

$$Z = \begin{bmatrix} | & | & & | \\ \sum_{j=1}^{n} a_{1j} V_j & \sum_{j=1}^{n} a_{2j} V_j & \cdots & \sum_{j=1}^{n} a_{nj} V_j \\ | & | & & | \end{bmatrix}^T$$

Since tokens naturally attend mostly to themselves, the value embedding of a token i itself is, as expected, most dominant in the result. However, tokens with key embeddings j matching the particular query embedding i yielding high attention scores a_{ij} additionally shape the result.

Cross-Attention. The concept of cross-attention works almost similarly mathematically but compares two different embedded sequences $X, Y \in \mathbb{R}^{n \times d}$. Here, the query embeddings Q are obtained from X by XW_q, the key and value embeddings K and V are obtained from Y by YW_k and YW_v respectively, which can be summarized in:

$$\text{Cross-Attention}\left(W_q, W_k, W_q, X, Y\right) := \text{Attention}\left(XW_q, YW_q, YW_v\right)$$

Final Remarks. It becomes evident why self-attention becomes first individually operated in $Enc_w\left(X\right)$ and $Dec_w\left(X\right)$. We need a token-matching com-

ponent, and pre-trained embeddings like GloVe provide a "multiplicative component," which is used for cosine similarity matching. Since token embeddings are added to the resulting embeddings in the linear combinations, it gains insight into why the "additive component" becomes so prominent in these embeddings. Concerning a specific token, high attention scores allow access to the information of value embeddings from other important related tokens.

2.5 Background Random Matrix Theory

Assume we have a matrix A of shape $n \times n$ with each element A_{ij} i.i.d normally distributed according to $\mathcal{N}\left(\mu,\, \sigma^2\right)$ with $\mu = 0$.

In simple terms, all exact points have a point probability of 0 in a randomized real-valued system; thus, all values and calculated results always differ. The distinctiveness of eigenvalues and singular values makes randomized matrices diagonalizable, and so there exists a diagonalizable decomposition $A = U\Lambda U^T$. Unfortunately, those eigenvalues can be complex numbers, which can be encountered by deploying the singular value decomposition $A = U\Lambda V^T$ instead, ensuring real diagonal matrix entries.

Non-quadratic Random Matrices. Let UDV^T be the singular value decomposition of A. This matrix A has almost surely full rank and, thus, all singular values $\sigma_1, \ldots, \sigma_n$ are distinct and nonzero. We focus on quadratic matrices since non-quadratic matrices of shape $n \times m$ with $n < m$ have still just $rank\,(A) = n$ nonzero eigenvalues, which results in ignoring the $m \times m$ sub-matrix of the $n \times m$ right singular value matrix V.

Random Vector Scaling Effects. Returning to the quadratic matrix A, we take an arbitrary unit vector v. The operations U and V as orthogonal matrices do not alter the vector length $||v||_2$; however, the operation Dv results in an approximated expected vector length of $\mathbb{E}[||Dv||_2] = n \cdot \sigma^2$. If $A_{ij} \sim \mathcal{N}\left(0,\, \frac{1}{n}\right)$, as commonly used for initializing weight matrices, it results in no undesired scaling effects. The output is expected to be again a unit vector, with all orientations in $R^{n \times n}$ being equally probable due to the randomness of the orthogonal rotations U and V.

Distribution of Squared Singular Values. The following sub-section has been supported by GPT-4, as well as the papers [38] and [24] as well as minor numerical experiments to test the postulated formulas for correctness.

The decisive key point is that we can imagine that the descendingly ordered singular value spectrum is fixed for a random matrix and a fixed marginal element i.i.d. distribution A_{ij}, and we sample from this spectrum with regular distance. However, these spot positions are slightly inaccurate and characterized by small uncertainties to the left and right, but on average, the distance is regular. The sampling becomes more precise for increasing matrix size n, and the singular value spectrum converges against a fixed pre-determinable function. The following Random Matrix Theory segment is summarized from [38, 24], their sources, and checked against GPT4.

If the matrix A is symmetrical $A = A^T$, then the Wigner semicircle law states that most singular values are distributed according to a Wigner semicircle pattern. However, non-symmetrical matrices often exhibit similar patterns. Regardless, for a $n \times n$ matrices, the Marchenko-Pastur law states that if n grows large, the probability density function for the squared singular values (absolute eigenvalues) converges to:

$$f(x) = \frac{1}{2\pi\sigma^2 x} \cdot \sqrt{4\sigma^2 x - x^2} \quad \forall x: \quad 0 \leq x \leq 4\sigma^2$$

If also, $\sigma = \frac{1}{\sqrt{n}}$, then the probability density function simplifies to:

$$f(x) = \frac{n}{2\pi x} \cdot \sqrt{\frac{4x}{n} - x^2} \quad \forall x: \quad 0 \leq x \leq \frac{4}{n}$$

Then the mean and the largest singular values alter according to the aspect ratio $\frac{m}{n}$ for asymmetric $n \times m$ matrices, which are encountered by the Xavier or Glorot weight initialization $\mathcal{N}(0, \frac{2}{n+m})$ [21]. Since we are mainly interested in symmetric matrices, Xavier initialization becomes $\mathcal{N}(0, \frac{1}{n})$, which we will frequently use in our theoretical analysis.

For the case of quadratic random matrices $n \times n$ with Xavier initialization [21], the singular value spectrum gets regularly sampled from the following function, depicted in Figure 2.4.

Diverging Singular Values As a small side note, the gaps of the singular value decomposition are defined as $\sigma_i - \sigma_{i-1}$ with $1 \leq i \leq n-1$, and empirically, it holds that the smallest gap $\min(\sigma_i - \sigma_{i-1})$ is proportional to σ and

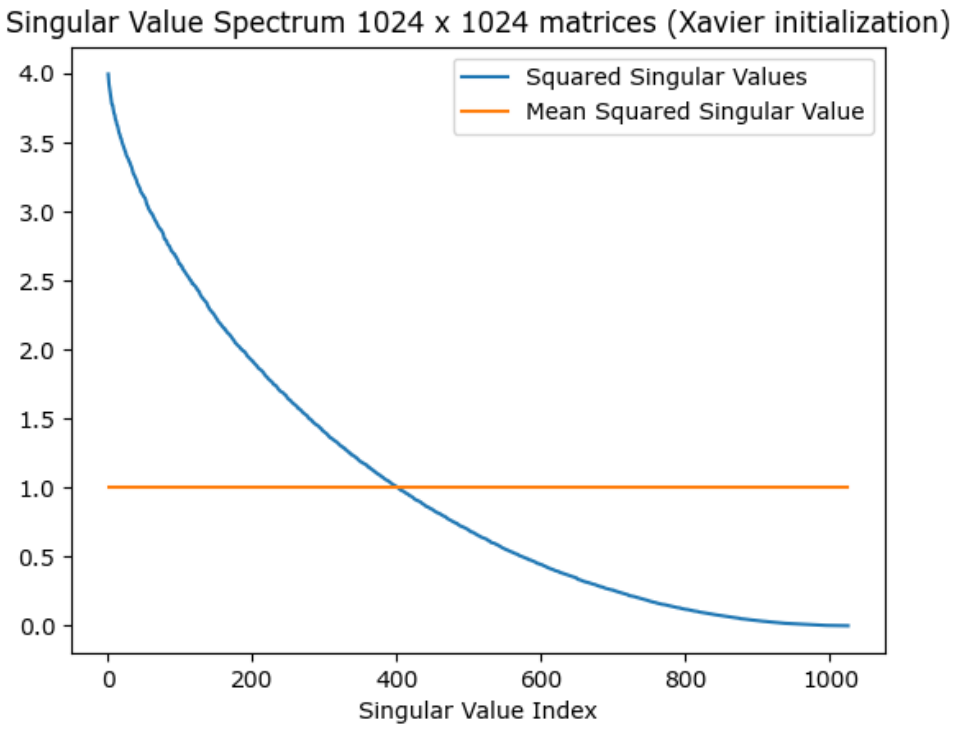


Fig. 2.4.: Singular Values for 1024×1024 matrices A with $A_{ij} \sim \mathcal{N}(0, \frac{1}{n})$

anti-proportional to the matrix size n so that $E[\min(\sigma_i - \sigma_{i-1})] \approx c \cdot \frac{\sigma}{n}$. This is surprisingly useful because the singular value decomposition samples from a fixed singular spectrum density function and very abstract repulsion effects ensure regularity of the sampling so that two spectra of two equally initialized random matrices follow very similar spectral properties with regard to scaling effects. Consequently, they can be interchanged, with the left and right singular value matrices predominantly identifying the matrix. Fascinatingly, it also gives an upper bound for the singular vector rotation under small perturbations, a key objective in Perturbation Theory.

3 Related Work

3.1 Transformer Attention Interpretability

Even though the novel field of Transformer [43] has catalyzed a sharp rise in research revolving around ubiquitous use cases and variants, the volume of work aimed at understanding the underlying mechanics by applying advanced mathematical techniques remains comparatively low. Hundreds of papers are adjusting attention or other model components to create a new transformer variant with better performance for a set of use cases [22]. In contrast, both the original and its derivative variants remain largely unexplained. Indeed, the branch of research dedicated to understanding these models' capabilities from a mathematical perspective is far more specific and requires delving into less well-known sources.

BERT Interpretability. In 2019, there was a wave of BERT attention explanation papers [8, 27, 20], which made structural empirical conclusions that specific language patterns are mirrored in the attention score maps from attention heads. Among these, the influential paper by Clark et al. [8] provides evidence that a general rule of thumb in NLP is that models learn syntactic dependencies in lower layers and semantic dependencies in higher layers. This finding is not particularly surprising, given that many language models, such as BERT, are pre-trained on tasks involving Part-of-Speech Tagging and Syntactic Tree Parsing. Consequently, syntactic sentence information becomes faintly visible in the patterns of the attention maps. This paper also discusses the concern that some (not all) attention heads show similarities, leading to the work of Paul Michel et al., *"Are Sixteen Heads Really Better than One?"* [27]. This study supports the observation of attention map pattern similarity but notes that attention heads focus on distinct language features even when grouped. Recalling the introduction, it is stated that the pre-trained embedding is split into attention heads, each processing a distinct part and conveying its own information.

Attention Interpretability. It should be noted that there has been an ongoing debate concerning the interpretability of attention patterns. One of the

seminal works that sparked this discussion is *"Is Attention Interpretable?"* [39], pointing out that the attention scores in terms of magnitude should not be over-interpreted and can often exhibit behavior that deviates from NLP researchers' expectations [29]. One aspect that was mentioned by Bing Bai et al. [3] is that tokens that are important for subsequent downstream tasks, like separator tokens, receive relatively great attention, which is explained by the fact that many language transformers are pre-trained on a variety of NLP tasks involving those special tokens, such as those in the GLUE benchmark [45].

3.2 Transformers and General Explainability & Interpretability

Gladly, there is newer research targeting the behavior of Attention $(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$ in more detail. There is ongoing research addressing the improvement of understanding of the internal dynamics of transformers, although this field is still in its nascent phase.

Explainable AI (XAI). This work differs somewhat from the main branch of Explainable AI (XAI) because it does not attempt to explain the model's classifications regarding which features or regions from the input space are most important to the model's decisions. Instead, it analyzes the mathematical processing of the Transformer model in a more generalized manner, as the analysis does not involve integrating gradient information. Nevertheless, XAI remains a valuable toolkit for gaining insights into the decision-making processes of Transformer models. Notably, a recent post-hoc explainability method by (Hila et al., 2021) [6] extends the concept of Layer-wise Relevance Propagation [5] to Transformer models. This model also overcomes the numerical instability issues arising from the skip connections in the transformer models. This model is one of the best and most competitive post-hoc explanation methods, surpassing many older XAI tools such as SHAP and GradCAM [25, 37], and many other explanation methods [35].

Weight Matrices Understandability. Another identifiable branch revolves around understanding the importance of neurons for the network's functioning. The debate about the *"Lottery Ticket Hypothesis"* [15] is the most prominent example, which extracted a small sub-network as performing the most work. This could go hand in hand with the idea of models indirectly

managing weighted linear combinations with respect to the latent space, yet unperceivable from the outside.

Vision Transformer Interpetability. Even though this work is not about Vision Transformers (ViT) [11, 23], they operate similarly to Language Transformers. For instance, an image of size $W \times H$ can be split up in $d_W * d_H$ patches of size $\frac{W}{d_W} \times \frac{H}{d_H}$. These patches can then be treated as 2-dimensional embeddings, comparable to tokenized embeddings in language models, and processed similarly, with the additional advantage of convolution due to the extra dimension. Paul et al. [30] demonstrated illustratively that different heads attend to other features in the spatial image domain; With this in mind, we can think of the attention map from the average head in one layer; it can be considered a weighted intermediate pattern, which we discuss later in our work. DeepViT [49] also addresses the problem of *"attention collapse"* where attention maps become more similar in deeper layers, akin to the *"token similarity problem"* [47].

3.3 Examining Transformers with Linear Algebra and Gaussian Distributions

Gaussians in Transformer Models. A recurring question is the role of Gaussian distributions as a supplementary tool in deep learning research. Recently, Gaussian distributions have been used to adjust the attention module in Transformer architectures [16]. Approaches to modeling the embedded languages via Gaussians are somewhat older [44, 28] and have become less prominent in recent research. Unfortunately, concerning the Central Limit Theorem through Layer Normalization [2], I have not found any work simulating the entire data pipeline in Transformer models with joint Gaussian distributions.

Linearity in Transformer Models. Another essential aspect of this work is the concept of linearity for transformers and how non-linearity offers opportunities to better understand the model's dynamics. Zhao et al. [48] analyze the geometric structure of intermediate embedding results and discover that they share characteristics with a cone or multiple overlapping cones even in high dimensional space. This aligns with our view of the embeddings located on a hypersphere and illustrates how the models could overcome the Curse of Dimensionality. If intermediate vectors are located within the bounds of

a cone rather than being chaotically structured, they are not always rectangular to themselves, as in a chaotic randomized system. That property also holds true for our pre-trained embeddings and tends to prevail in intermediate representations. The representation power was recently analyzed in *"Representational Strengths and Limitations of Transformers"* [36].

Singular Value Spectrum in Transformer Models. The singular value or eigenvalue domain is crucial for the attention dynamics, as analyzed by Bhojanapalli et al. [4]. They discovered that attention scores are distributed in a low-rank space. Dynamic Bilinear Low-Rank Attention (DBA) [32] and LightFormer [26] utilize the factor that only a few eigenvalues (singular values) are of high magnitude to approximate computations.

Linearizability of Transformer & Euclidean Distance Preservation. In 2020, a seminal paper inventing *"Linformer"* [46] shaped linearized attention and provided rigorous proof that attention is low-rank and maintains distances of points by applying the John Lindenstrauss Lemma. This is significant because it not only accelerates computations by decreasing runtime complexity from $\mathcal{O}\left(n^2\right)$ to $\mathcal{O}\left(n\right)$ but also reveals a small, nearly linear bottleneck (sparse singular value spectrum) that determines most of the structure of the resulting attention map.

Bilinear form as Gaussian Kernel. Moreover, a non-negligible branch for the analysis of attention is viewing it as a kernel function [42], structurally related to RBF kernels, though in our case, it is a non-symmetric kernel for the non-symmetric standard self-attention mechanism as non-symmetric bilinear form.

Linearizability of Transformer & Euclidean Distance Prservation. One advanced problem that is caused by sparse singular value spectra from the attention weight matrices W_q and W_k is the from Hanqi Ya et al. unveiled *"token similarity problem"* [47], with all embedding distributions strongly resembling each other after multiple attention passes in higher layers, which we discuss further towards the end of this work.

3.4 Transformers Advanced Interpretability

Communication Channel & Polysemanticity. A seminal branch of research in this area has been developed by the Anthropic's Interpretability Research

Group, which laid the groundwork for an advanced interpretation of Transformer models. They formulated an entire mathematical framework for Transformers, introducing the concept of the residual stream as a communication channel through skip connections [13]. Moreover, they noted that many linear structures could be applied to Transformer models, justifying using both Gaussians and linear combinations as explanatory methods. The most significant research objective is polysemanticity [12], which describes how the model represents features in a base with more parameters than the activation space (corresponding to embedding dimensionality), which are then compressed while preserving geometry (as proposed by the *"Linformer"* paper [46]). The multitude of representation power ties with the mixture of probability distributions and their associated information content. Examining this issue unquestionably marks a milestone in the process of demystifying Transformer Models.

4 Data Distribution under Gaussian Approximations

4.1 Gaussian Modelling of the Input Space

This chapter starts to develop the core of this work, the mathematical model describing the statistical distribution of the data, which our model learns in terms of Gaussian Distributions. For the rest of the theoretical investigation, we aim to track the statistical moments of the Multivariate Normal Distribution throughout an attention module forward pass.

4.1.1 Definitons Tokenizer and Embeddings

We begin with a transformer model $\{Enc_w(x), Dec_w(x)\}$ and a language dataset $D = \{S^{(j)}\}$ consisting of m sentences $S^{(1)}, .., S^{(m)}$. Each sentence $S^{(j)}$ is tokenized by a model tokenizer T_M, which maps each word to a discrete number in $[0, V]$, where $V \in \mathbb{N}$ is the vocabulary size. All sentences whose tokenized length is smaller than that of the longest sentence are padded with dummy tokens, ensuring all tokenized sentences have the same length n.

Then each token $t_i \in [0, 1, .., V-1]$ is embedded by the model's embedding layer Emb of the encoder $Enc_w(x)$ or decoder $Dec_w(x)$, which assigns each token t_i a d-dimensional embedding vector $emb(t_i) \in \mathbb{R}$. Theoretically, there are only V different embedding vectors, but V is often large (> 30000), and at the very least, "positional encodings" are added to the embedded tokens, so there is sufficient variety given.

An arbitrary sequence of n embedded tokens: $emb(t_1), \ldots, emb(t_n)$ are collected together in the matrix $X = \begin{bmatrix} X_1^{(j)} & X_2^{(j)} & \ldots & X_n^{(j)} \end{bmatrix} \in \mathbb{R}^{d \times n}$, so that $\forall X_i : X_i \in \mathbb{R}^d$. We now refer to X as a sequence of n embeddings of dimensionality d for an arbitrary sentence j, which we omit in our notation.

4.1.2 Fitting a large Joint Gaussian

How can we best model the probability distribution of the data sequence $X \in \mathbb{R}^{d \times n}$? We assume that each embedding $X_i \in \mathbb{R}^d$ of a token t_i follows a Gaussian distribution $p_i : \mathcal{N}_i(\mu_i, \Sigma_i)$, with $\mu_i \in \mathbb{R}^d$ and $\Sigma_i \in \mathbb{R}^{d \times d}$. Now, we stack all Gaussians together to form an even larger Gaussian and model a sentence with a large joint Gaussian of dimension $n \cdot d$. The combined Gaussian $\mathcal{N}(\mu, \Sigma)$ with $\mu \in \mathbb{R}^{n \cdot d}$ and $\mu \in \mathbb{R}^{(n \cdot d) \times (n \cdot d)}$, obtainable via Principal Component Analysis (PCA) as described in Section 2.1, for a sentence would then look like this:

$$\mathcal{N}(\mu_X, \Sigma_X) \qquad \mu_X = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_n \end{bmatrix} \quad \Sigma_X = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{21} & \dots & \Sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{n1} & \dots & \dots & \Sigma_{nn} \end{bmatrix} \tag{4.1}$$

The new joint Gaussian matrix Σ_X is indeed positive definite with positive singular values and $\forall i, j : \Sigma_{ij} = \Sigma_{ji}^T$ due to the definition of the (cross-)covariance matrix.

4.2 Examining Characteristics of Joint Gaussians

The Gaussian distribution Σ can be understood as a joint probability distribution of $(X_1, \dots, X_n)$ with individual covariance matrices $\Sigma_{ii} \in \mathbb{R}^{d \times d}$ as marginal probability densities for the variables $X_i \in \mathbb{R}^d$ with mean $\mu_i \in \mathbb{R}^d$. The cross-covariance matrices $\Sigma_{ij} \in \mathbb{R}^{d \times d}$ model the interdependence between the variables X_i and X_j and thus represent the dependence of words from the viewpoint of the human language. We also refer to the individual Gaussian distributions $\mathcal{N}_i$ with p_i. From a stochastic perspective, we have treated $X_1, X_2, \dots, X_n$ as a multivariate finite stochastic process of length n so that a cross-covariance matrix Σ_{ij} captures the statistical dependence between timesteps i and j.

There are a lot of statistical properties to discuss based on that approach:

- Every marginal distribution is a Gaussian, applying not only to the distributions $\mathcal{N}_i$, but also to each feature component d_k in $X_i \in \mathbb{R}^d$ is modeled by a univariate Gaussian $\mathcal{N}_{ik}(\mu_{ik}, \sigma_{ik})$, making the distribution symmetric along each feature dimension.

- The Mutual Information $I(X_i; X_j)$ between X_i and X_j is driven by the determinant of their marginal covariance matrices Σ_{ii} and Σ_{jj}, distorted by their respective cross-covariance matrix Σ_{ij}.

- We can also assume that words with greater distances in a sentence tend to impact each other on average to a lesser extent. This implies that increasing $|i-j|$ would result in the determinant of Σ_{ij} becoming smaller, and their Mutual Information $I(X_i; X_j)$ decreases accordingly.

4.2.1 Joint Gaussian Distributions under Linear Maps

One of the primary motivators advocating the choice of Multivariate Gaussian is the relative simplicity of transforming these distributions under linear maps. If $L \in \mathbb{R}^{d \times d}$ is an invertible linear map, which is applied to each embedded token X_i of our sentence, then the new distribution associated to $LX_i \in \mathbb{R}^d$can be modeled as:

$$\mathcal{N}(\mu_{LX}, \Sigma_{LX}) \quad \mu_{LX} = \begin{bmatrix} L\mu_1 \\ L\mu_2 \\ \dots \\ L\mu_n \end{bmatrix} \quad \Sigma_{LX} = \begin{bmatrix} L\Sigma_{11}L^T & L\Sigma_{12}L^T & \dots & L\Sigma_{1n}L^T \\ L\Sigma_{21}L^T & L\Sigma_{21}L^T & \dots & L\Sigma_{2n}L^T \\ \vdots & \vdots & \ddots & \vdots \\ L\Sigma_{n1}L^T & \dots & \dots & L\Sigma_{nn}L^T \end{bmatrix} \tag{4.2}$$

4.2.2 Information Theory on Transformed Gaussians

We delve into the question how Information Theory applies to our cumulative joint Gaussian distribution.

Entropy of Transformed Gaussians. If the invertible linear map L has a singular value spectrum with a deforming volume $\det(L) \neq 1$, it shapes the entropy of each new marginal Gaussian for each $\hat{X}_i = LX_i$ with $\hat{X}_i \sim \mathcal{N}\left(L\mu_i, L\Sigma_{ii}L^T\right)$:

$$\mathrm{H}\left(\hat{X}_i\right) = \frac{1}{2}\log\left((2\pi e)^d \left|L\Sigma_{ii}L^T\right|\right) = \frac{1}{2}\log\left((2\pi e)^d \det(L)^2 \det(\Sigma_{ii})\right) =$$
$$\frac{1}{2}\log\left((2\pi e)^d \det(\Sigma_{ii})\right) + \frac{1}{2}\log\left(\det(L)^2\right) = \mathrm{H}(X_i) + \log\left(|\det(L)|\right)$$

We can also conclude, that if $\sigma_1, \ldots, \sigma_n$ are the singular values of L, then the entropy changes by $\log\left(|\det(L)|\right) = \log\left(\prod_{i=1}^{n} \sigma_i\right) = \sum_{i=1}^{n} \log \sigma_i$

If the product of singular values of L is greater than 1, indicating $\det(L) > 1$, the entropy increases, reflecting growing uncertainty. Conversely, if the product of squared singular value of L is less than 1, both entropy and uncertainty decrease. Hence, if the model scales down the weight matrices in linear layers, it transforms an indefinite chaotic probability distribution into a more meaningful and ordered configuration.

Mutual Information of Transformed Gaussians. Concerning the Mutual Information $\mathrm{I}\left(\hat{X}_i; \hat{X}_j\right) = \mathrm{I}\left(LX_i; LX_j\right)$, the new determinant expression results in:

$$\mathrm{I}\left(\hat{X}_i; \hat{X}_j\right) = \frac{1}{2} \log \frac{|\Sigma_{LX_i}||\Sigma_{LX_j}|}{|\Sigma_L|}, \quad \Sigma_L = \begin{bmatrix} L\Sigma_{ii}L^T & L\Sigma_{ij}L^T \\ L\Sigma_{ji}L^T & L\Sigma_{jj}L^T \end{bmatrix}$$

Applying properties of block matrices, it follows that $\det(\Sigma_L) = \det(L)^4 \cdot \det(\Sigma)$ and thus volume change factor $\det(L)$ can be canceled out in both the nominator and denominator:

$$\mathrm{I}\left(\hat{X}_i; \hat{X}_j\right) = \frac{1}{2} \log \frac{|\Sigma_{LX_i}||\Sigma_{LX_j}|}{|\Sigma_L|} = \frac{1}{2} \log \frac{\det(L)^2 |\Sigma_{ii}| \det(L)^2 |\Sigma_{jj}|}{\det(L)^4 |\Sigma|} = \mathrm{I}\left(X_i; X_j\right)$$

That implies that statistical mutual interdependencies are consistently modeled through linear layers; only non-linear transformations alter them, making Mutual Information a powerful qualitative measure.

The same linear invariance can also be shown for the KL divergence in the case of Gaussian variables, where, effectively, the linear map L cancels out in each individual term.

4.2.3 Bayesian Preconditioning

Applying this Gaussian model directly to raw embedded text data has several drawbacks; most notably, there are too many words a sentence could start with, each with different orientations on the hyper-sphere, rendering the Gaussian distributions for each token less useful.

Therefore, it becomes reasonable to incorporate Bayesian statistics by instead analyzing $P(X_{i+1}|X_i, \dots, X_1) \approx P(X_{i+1}|X_i)$ into that procedure using a word co-occurrence matrix format in a 1-gram model format. The entries could be estimated from the transformation matrix via a Markov Chain utilized on a word co-occurrence matrix in a 1-gram model format.

The huge benefit is that a few words could dominate the probability distribution then, making the estimated Gaussian distribution more meaningful. For instance, if token t_i is followed by any of the five tokens $t_1, \dots, t_5$ with an overall probability exceeding 0.9, then the Gaussian $p_i|p_{i-1}$ is dominated by the corresponding five embedding vectors. In this representative format, the remaining tokens are included by a noise term ϵ, resulting in a weighted linear combination of embedding vectors as a parametrization of $p_i|p_{i-1}$.

4.3 Spectral Properties of Marginal Gaussians

We now try to embed the prior knowledge from language embeddings into the joint Gaussian model, concretizing the marginal Gaussian distribution and allowing us to draw conclusions about the singular value spectrum and the rank of the covariance matrix.

4.3.1 Singular Value Spectrum of Marginal Gaussian

We now focus on the marginal token probability distribution associated with X_i and its characteristics. Concretely, we have k probable tokens $v_1, \dots, v_k \in \mathbb{R}^d$ with respective probability scores $\lambda_1, \dots, \lambda_k$, such that $\sum_{i=1}^{k} \lambda_i = 1$. This distribution can be discretized by vertically stacking these vectors by their frequency into a matrix Y. Considering the unnormalized covariance matrix YY^T, this is equivalent to stacking the vectors vertically, scaled by their respective probability scores λ_i. We can now think about the covariance matrix as a weighted linear combination of outer vector products $\sum_{i=1}^{k} (\lambda_i \cdot v_i) \cdot (\lambda_i \cdot v_i)^T = \sum_{i=1}^{k} \lambda_i^2 \cdot (v_i \cdot v_i)^T$.

We know the inner product Y^TY because it contains the angular information, or more formally, the pair-wise scalar products α_{ij} of the pre-trained embeddings $\langle v_i, v_j \rangle = \langle v_j, v_i \rangle = \alpha_{ji} = \alpha_{ij}$. Most embedding vectors are of the same length l or, better said, lie on a hypersphere of length l.

We simplify calculations by setting $l = 1$, leading to $\alpha_{ii} = 1$. Gaussians in high dimensions have some characteristics resembling a hypersphere, making the Gaussian an appropriate choice as probabilistic parametrization for this constrained data manifold. The outer product matrix $YY^T = \sum_{i=1}^{k} \lambda_i^k v_i v_i^T$ has at most rank k, thus k nonzero singular values and we conclude for the marginal covariance matrix Σ_{ii} of X_i:

$$X_i \approx YY^T = \sum_{i=1}^{k} \lambda_i v_i v_i^T, \quad \sum_{i=1}^{k} \lambda_i = 1 \qquad \Sigma_{ii} = U \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & \sigma_d \approx 0 \end{bmatrix} V^T$$

Although the noise terms generate nonzero singular values, they are negligible compared to the singular values $\sigma_1, \dots, \sigma_k$. To avoid confusion and clearly establish the terminology, speaking of sparse singular value spectra implies that most singular values are infinitesimal close to zero. For a continuous real-valued system, every point has a point probability of 0, so the point zero does not exist in an abstract sense.

From Linear Algebra theory, it can be derived that the largest k absolute eigenvalues of the inner product matrix Y^TY or the outer product matrix YY^T correspond to the k largest squared singular values of the diagonal matrix in the singular value decomposition $Y = UDV^T$, providing us the determinant or volume of the resulting Gaussian. Therefore, we use the Gram matrix Y^TY as it allows us to describe it by our pre-trained angular information and obtain its eigenvalues. Reformulating the matrix would look like this:

$$Y^TY = \begin{bmatrix} \langle \lambda_1 v_1, \lambda_1 v_1 \rangle & \langle \lambda_1 v_1, \lambda_2 v_2 \rangle & \dots & \langle \lambda_1 v_1, \lambda_k v_k \rangle \\ \langle \lambda_2 v_2, \lambda_1 v_1 \rangle & \langle \lambda_2 v_2, \lambda_2 v_2 \rangle & \dots & \langle \lambda_2 v_2, \lambda_k v_k \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \lambda_k v_k, \lambda_1 v_1 \rangle & \dots & \dots & \langle \lambda_k v_k, \lambda_k v_k \rangle \end{bmatrix} \approx \begin{bmatrix} \lambda_1^2 & \lambda_1 \lambda_2 \alpha_{12} & \dots & \lambda_1 \lambda_k \alpha_{1k} \\ \lambda_1 \lambda_2 \alpha_{12} & \lambda_2^2 & \dots & \lambda_2 \lambda_k \alpha_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1 \lambda_k \alpha_{1k} & \dots & \dots & \lambda_k^2 \end{bmatrix}$$

The trace of Y^TY is obviously $||\lambda||_2 = \sum_{i=1}^{k} \lambda_i^2$; the determinant is more complicated as $\prod_{i=1}^{k} \sigma_i$ is dependant on how α_{ij} are distributed and is not simply calculable.

4.3.2 Angular Information & and Diverging Singular Values

Calculating the exact distribution of the eigenvalues/singular values is unquestionably challenging because it requires preconditioning the distribution of α_{ij}, which is expected to behave asymmetrically since learned embeddings tend still towards orthogonality, skewing the distribution of α_{ij}, and so does Wigner's semicircle law apply not so well.

Nevertheless, we can briefly glance at that problem by focusing on v_i and v_j, assuming all other vectors are orthogonal. Without loss of generality, let these vectors be v_1 and v_2:

$$Y^T Y \approx \begin{bmatrix} \lambda_1^2 & \lambda_1\lambda_2\alpha_{12} & \dots & 0 \\ \lambda_1\lambda_2\alpha_{12} & \lambda_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_k^2 \end{bmatrix}$$

The eigenvalues of the submatrix $\begin{bmatrix} \lambda_1^2 & \lambda_1\lambda_2\alpha_{12} \\ \lambda_1\lambda_2\alpha_{12} & \lambda_2^2 \end{bmatrix}$ are given by:

$$\lambda_{1,2} = \frac{\lambda_1^2 + \lambda_2^2 \pm \sqrt{\left(\lambda_1^2 + \lambda_2^2\right)^2 - 4\lambda_1^2\lambda_2^2\left(1 - \alpha_{12}^2\right)}}{2}$$

This form may appear complex, but if $\lambda_1 \approx \lambda_2 = \lambda$, then $\lambda_{1,2} \approx \lambda\left(1 \pm \alpha_{12}\right)$, elucidating higher scalar product ties with diverging eigenvalues stretched by α_{12}. In fact, average higher scalar products α_{ij} go along with repelling singular values, making the singular value spectrum less balanced and thus more sparse, even though the exact deformation is difficult to determine.

4.4 Summary Data Distribution under Gaussian Approximations

To wrap up the theoretical findings, we have demonstrated that Gaussian distribution serves as a mathematically valid starting point when compressed into a global probability distribution matrix. The localization of embeddings on a hypersphere arises from the consistent lengths among the pre-trained embeddings. We have also analyzed how linear maps and so linear weight

layers affect Gaussian statistics with regard to their spectral properties, showing how the entropy becomes altered and that mutual interdependencies are well-preserved under linear operations. Furthermore, by pre-conditioning the joint Gaussian propelled from a Bayesian perspective, we can estimate the spectral properties of the Gaussian and fathom how they got affected by prior angular information or the Gram Matrix of the pre-trained embeddings. This restriction in terms of Gaussian statistics becomes practical when operating on the non-linear parts of the model, which is the main topic of the next section.

5 Statistics and Geometry of Softmax in Attention

5.1 The Softmax Function: Differentiable and non-linear

This section delves into the mathematical nature of the attention mechanism, particularly focusing on the nonlinear Softmax operation.

5.1.1 Definition of Probability Spaces in Softmax

The Softmax transformation encompasses two fundamentally different parts, each individually operated on a row to translate into a probability space. First, we should define a probability space for a row i. The matrix entry A_{ij} corresponds to the event (column) j in row i occurring with probability A_{ij}. More formally, the probability space for row i gets defined as $P_i = \left(\Omega, \mathcal{F}, \hat{P}_i\right)$ with $\Omega = [1, 2, \ldots, n]$, $\mathcal{F} = 2^{\Omega}$, and probability measure $\forall A \subseteq \Omega : \hat{P}_i(A) = \sum_{j \in A} A_{ij}$. We define the row variables $X_1 \sim P_1, X_2 \sim P_2, \ldots, X_n \sim P_n$.

It is essential to break down the Softmax function in more detail:

The Softmax function consists of element-wise exponentiation followed by normalization, resulting in

$$\left[\frac{\exp(x_1)}{\sum_{i=1}^{n}\exp(x_i)} \quad \frac{\exp(x_2)}{\sum_{i=1}^{n}\exp(x_i)} \cdots \frac{\exp(x_n)}{\sum_{i=1}^{n}\exp(x_i)}\right]$$

, completing the transformation. We denote $\hat{A}$ as the matrix after the exponentiation and $\bar{A}$ as the matrix after the consecutive normalization. Similarly, let $\hat{X}_1, \hat{X}_2, \ldots$ and $\hat{X}_n$ denote the random row variables of an arbitrary row of $\hat{A}$ and $\bar{X}_1, \bar{X}_2, \ldots$ and $\bar{X}_n$ that of $\bar{A}$.

5.1.2 Softmax Part One: Exponentiation

Starting with the exponentiation of the matrix elements $\forall (i,j) \in [1,n] \times [1,n] : \hat{A_{ij}} = \exp(A_{ij})$, the transformation does obviously not preserve the eigenstructure of the matrix A due to its non-linearity, requiring sophisticated mathematical modeling. That necessitates preconditioning the distribution of the input values to a normally distributed one. We will later clarify why this condition still seems plausible. Then $\hat{X}_i$ is log-normally distributed with the following statistical moments:

$$X \in \mathcal{N}(\mu, \sigma^2), \quad \mathbb{E}[\exp(X)] = \exp\left(\frac{\mu + \sigma^2}{2}\right) \quad \mathrm{Var}(\exp(X)) = \left(\exp(\sigma^2) - 1\right) \cdot \exp(2\mu + \sigma^2)$$

Under Xavier initialization $x = \begin{bmatrix} x_1 \dots x_n \end{bmatrix} \sim \mathcal{N}(0, diag\{\frac{1}{n}, \dots, \frac{1}{n}\})$ and using Taylor approximation of order 2 yields for the value domain of the log-normal distributed variable:

$$\mathbb{E}[exp(x)] \approx \begin{bmatrix} \exp(\frac{1}{2n}) & \dots & \exp(\frac{1}{2n}) \end{bmatrix}^T \approx \begin{bmatrix} 1 + \frac{1}{2n} + \frac{1}{8n^2} & \dots & 1 + \frac{l}{2n} + \frac{l^2}{8n^2} \end{bmatrix}^T \tag{5.1}$$

$$\mathrm{Var}(exp(x)) \approx \begin{bmatrix} \exp(\frac{2}{n}) - \exp(\frac{1}{n}) & \dots & \exp(\frac{2}{n}) - \exp(\frac{1}{n}) \end{bmatrix}^T \approx \begin{bmatrix} \frac{1}{n} + \frac{3}{2n^2} & \dots & \frac{1}{n} + \frac{3}{2n^2} \end{bmatrix}^T \tag{5.2}$$

We can observe that $\mathbb{E}[\hat{X}_i] \approx 1 + \frac{1}{m}$ and $\mathrm{Var}\left(\hat{X}_i\right) \approx \frac{1}{n}$ Moreover, the exponentiation is invertible and bijective, indicating no information gain concerning this operation.

$$\mathbb{E}[exp(x_i)] \approx 1 + \frac{1}{2n} + \frac{1}{8n^2} \qquad \mathrm{Var}(exp(x_i)) \approx \frac{1}{n} + \frac{3}{2n^2}$$

5.1.3 Softmax Part Two: Normalization

Now we address the intricate normalization of each row matrix, which resembles multiplying each row by a scalar $\alpha_1, \dots, \alpha_n$. So $A' = DA$, with $D = diag\{\alpha_1, \alpha_2, \dots, \alpha_n\}$, unfortunately, this scaling factor distribution is very complex. Even when $\hat{X}_i$ are log-normal distributed, their sum is not and can be estimated by a skewed Fenton-Wilkinson approximation [17]. This non-invertible transformation also disrubs the singular value structure of the matrix A'. While one might say that it just scales the rows, all scaling factors α_i are non-linearly dependent on each X_i via $\exp$. Consequently, this system eliminates one degree of freedom and restrains the points onto a simplex,

complicating statistical modeling. Nevertheless, the marginal distribution of $\bar{X}_i$ is a special case of the Boltzmann distribution, whose moments lack closed forms, but the relational dependencies $\frac{\hat{X}_i}{\hat{X}_j}$ are untouched by the scaling and can be analyzed.

However, this work aims to give meaning to the transformation's geometry and to provide a better understanding of how the model learns target probability distributions.

5.2 Linearity and Geometry of Softmax

5.2.1 Softmax Differentiation with Cross-Entropy & Linear Combinations

The Softmax operation in a self-attention layer is related to the final Softmax operation in the output layer of standard classification models. Here, $\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$ denote the input to the SoftMax function. Suppose we want to learn a discrete target distribution $q = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}$ for that row and compare it against the actual row probability distribution $p = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix}$. Minimizing the discrete cross-entropy $\mathrm{H}(q_i, p_i)$ for that case, where it is important that we fix the base of the logarithm, or the information unit to e instead of the Shannon base 2:

$$\mathrm{H}(q,p) = -\sum_{j=1}^{n} q_j \cdot \log(p_j), \quad p_j = \frac{e^{x_j}}{\sum_{k=1}^{n} e^{x_k}}$$

Combining the partial derivatives $\frac{\mathrm{dH}}{\mathrm{d}p_j}$ and $\frac{\mathrm{d}p_j}{\mathrm{d}x_j}$ with the product rule $\frac{\mathrm{d}H}{\mathrm{d}x_j} = \frac{\mathrm{dH}}{\mathrm{d}p_j} \cdot \frac{\mathrm{d}p_j}{\mathrm{d}x_j}$, we find:

$$\frac{\mathrm{dH(q,p)}}{\mathrm{d}x_j} = p_j - q_j$$

Learning probability distributions in an attention layer yields an error term concerning the probability terms. If we express p_i with a $p_i = \lambda_1 \cdot p_i^{(1)} + \cdots + \lambda_k \cdot p_i^{(k)}$, with $\sum_{i=1}^{k} \lambda_i = 1$, we obtain:

$$\frac{\mathrm{dH}}{\mathrm{d}x_j} = \sum_{l=1}^{k} \lambda_l \cdot \left(p_j^{(l)} - q_j\right)$$

It is worth mentioning that equally balanced linear combinations would result in gradients being shared amongst the backpropagation paths to the same magnitude. Notably, the task of classification (learning probability distributions) is structurally equivalent to derivating a quadratic regression term. While the value domain for regression tasks differs, the gradient remains proportional to the exponentiated error term between the estimated and true target.

Derivation illustrates a direct correspondence of linear combinations under differentiation. However, we want to establish a geometric connection independently to match vectors with their probabilistic Softmax counterparts. The key difference is that we deal with discrete probabilities after applying SoftMax, which we can classify and compare using information theory. This necessitates establishing a connection from the perspective of Euclidean geometry.

5.2.2 Geometry of Probability Spaces

The idea lies in interpreting each vector $v \in \mathbb{R}^d$ as a discrete probability distribution with n events, which would be obtained after applying Softmax and defining the probability space as stated above in section 5.1.1. Let $\hat{v} = \text{Softmax}\,(v)$, so that for the individual components, it holds $\sum_i^n \hat{v}_i = 1$ and event i occurs with probability $\text{Softmax}\,(v)_i$. The zero vector obviously has no corresponding probability distribution, but for a continuous random variable $v \in \mathbb{R}^d$, each particular event has a probability density value of zero, so we can simply ignore this case.

Next, we want to address the Softmax operation's normalization, eliminating one degree of freedom. The main analogous algorithm would be projecting each point onto the surface of a $n-1$-Sphere $\mathcal{S}_l^{n-1}$ with radius $l > 0$ via division through the Euclidean norm $||v||_2$, and then scaling by l. The zero vector gets projected to the point of infinity ∞, as typical in Projective Geometry. Thus, we can assign each discrete probability a vector and vice versa on the sphere. On the contrary, all discrete probability vectors are characterized by the Manhattan norm $||v||_1 = 1$, fixing these points to a n-dimensional

simplex's surface, casting doubt on the complexity of the transformation under exponentiation.

The set of discrete probability functionals $P : [1, .., n] \to [0, 1]$, containing all probability functions has the characteristic that all linear combinations of discrete probability density functions: $p(x) = \sum_{i=1}^{l} \lambda_i \cdot p_i(x)$ with $\sum_{i=1}^{l} \lambda_i = 1$ are again a probability function; however, multiplying $p(x)$ with a scalar $\alpha \neq 1 \in \mathbb{R}$ is not allowed, as the scaled result not a valid probability function, a missing component for a vector space.

This characteristic is also mirrored on the sphere as we can create a linear combination of unit vectors, but not by simple addition due to the nature of the Euclidean norm. Due to $||v \pm w||^2 = ||v||^2 + ||w||^2 \pm 2v^T w$, lengths are not automatically preserved, as they get rescaled by their dot product or their respective angle $v \cdot w \sim \cos(\angle(v, w))$.

Noteworthy, if two vectors v, w were randomized with Xavier initialization $v, w \sim \mathcal{N}\left(0, \frac{1}{n}\right)$, then:

$$\mathbb{E}[\langle v, w \rangle] = 0 \qquad \text{Var}(\langle v, w \rangle) = \frac{1}{n} \tag{5.3}$$

On the one hand, random vectors are expected to be rectangular, oscillating with relative stability. On the other hand, added random vectors are expected to preserve length likewise. This statement can be simply proven by aggregating n squared normal variables.

5.2.3 Connection between Linear Operations in Euclidean Geometry and Probabilistic Spaces

Linear Combinations under Softmax. We want to investigate how linear combinations $\sum_{i=1}^{n} \lambda_i \cdot v_i$ in the familiar Euclidean vector space $\mathbb{R}^d$ relate linear combinations of discrete probability distributions of their Softmax-transformed counterparts $\sum_{i=1}^{n} \hat{\lambda}_i \cdot p_i = \sum_{i=1}^{n} \hat{\lambda}_i \cdot \text{Softmax}(v_i)$ in the space of discrete probability functions P. Every linear combination $\lambda_1 + \lambda_2 + \cdots + \lambda_n$ can be divided by the euclidean norm $||\lambda||_2 = \sum \lambda_i^2$, so that $||\sum \frac{\lambda_i}{||\lambda||}||_2 = 1$. Therefore, we must analyze two cases: any linear combination with $||\lambda||_2 = 1$ and multiplication by a real scalar $\alpha||\sum \frac{\lambda_i}{||\lambda||}||_2 \in \mathbb{R} \setminus 0$. Concerning the latter case:

Softmax Vector Multiplication. First, we address the scaling of vectors $\alpha \neq 1$, which has no vector space compatible equivalent in discrete probability spaces. Well, probability distributions can be measured by entropy, a scalable component. Multiplying the vector with a scalar $\alpha > 1$ and subsequent exponentiation changes the orientation of the vector regarding the Euclidean norm $||\cdot||_2$ non-linearly, so does the entropy measure of the Softmax-transformed vector alike.

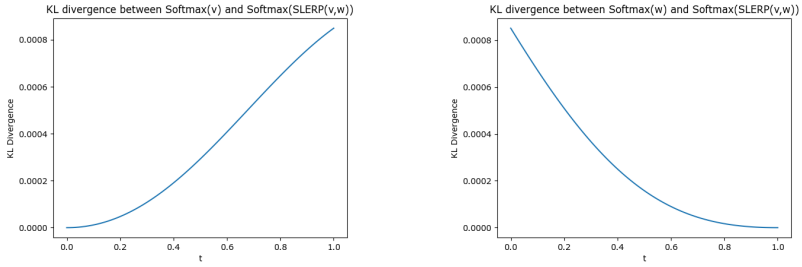
Each random real-valued vector is characterized by distinct elements, so let i be the index of its largest element $\max(v_i)$. The exponential growth of the largest element v_i makes it easy to prove that for $\alpha \to \infty$, $\text{Softmax}(\alpha \cdot v) \to \begin{bmatrix} 0 & \dots & 1 & \dots & 0 \end{bmatrix}^T$ approaches the Dirac delta measure δ_i with zero entropy. Regarding the model's capabilities, upscaling causes the probability distribution to converge to the Dirac delta measure of its largest element, and the distribution gets exponentially peakier.

On the contrary, downscaling with $\alpha < 1$ approaches a balanced state with maximum entropy $\begin{bmatrix} \frac{1}{n} & \dots & \frac{1}{n} & \dots & \frac{1}{n} \end{bmatrix}^T$, reaching the maximal entropy of $\sum_{i=1}^{n} -\frac{1}{n} \log\left(\frac{1}{n}\right) = \log(n)$.

Labeling the discrete probability vectors might misuse the term Dirac delta; however, I consider them a discretized equivalent to a continuous Dirac delta measure. Unfortunately, that makes geometrical comprehension challenging. Downscaling pushes all points to the same location in the discrete probability space, while upscaling vectors break linear combinations even further due to instability introduced by Dirac delta measures.

Softmax Vector Addition. We now have to just consider linear combinations $\lambda_1, \dots, \lambda_n$ with $||\lambda||_2 = 1$, as we already discussed multiplication. SLERP [14] spherically interpolates vectors on the surface so that $||\lambda||_2 = 1$ is fulfilled. Generalization to more than two vectors is technically too challenging, so we limit ourselves to two vectors v and w of unit length. They span a circle on the hypersphere's surface, with the SLERP interpolation walking along this circle's shortest path:

$$\theta = \angle(v, w) \quad \lambda_1(t) = \frac{\sin((t-1)\cdot\theta)}{\sin(\theta)} \quad \lambda_2(t) = \frac{\sin(t\cdot\theta)}{sin(\theta)}$$
$$SLERP(v_1, v_2, t) = \lambda_1(t)\cdot v1 + \lambda_2(t)\cdot v2 \qquad 0 \leq t \leq 1$$



(a) $(Sm\,(v)\,||Sm\,(SLERP\,(v,w,t)))$ **(b)** $(Sm\,(w)\,||Sm\,(SLERP\,(v,w,t)))$

Fig. 5.1.: KL divergence, averaged over 1000 passes of orthogonal unit vector pairs

After internalizing this concept, we can see that the probability distributions represented by Softmax (v_1) and $Softmax\,(v_2)$ get interpolated, and the intermediate distribution is given by Softmax $(SLERP\,(v_1, v2, t))$.

5.2.4 SLERP: The bridge between Euclidean Geometry and Probabilistic Spaces

My experiments indicated that for two random vectors v and w of length l, the KL divergence $D_{KL}\,(\text{Softmax}\,(w)\,||\text{Softmax}\,(SLERP\,(v,w,t)))$ followed a sinusoidal curve for most of the domain $0 \leq t \leq 1$ for a sphere's radius of $l = 1$, which is illustrated for orthogonal vectors in Figure 5.1, which are the objective of the upcoming proof. Finally, I also observed the same behavior for vectors of smaller vector length $l < 1$. Conversely, for $l > 1$ shifted the curve to a Dirac delta measure.

That sinusoidal behavior would imply that concerning the relational information $\theta = \angle(v,w) = \arccos\left(\frac{v^T w}{||v||_2||w||_2}\right)$, we get a direct reparametrization between angular information in the Euclidean space and the KL divergence information distance measure in the domain of discrete probability spaces. The proof is performed for rectangular vectors; arbitrary randomized vectors behave due to property 5.3 oftentimes in this regard, and this behavior can be observed approximately either.

Conjecture: Let $v, w \in \mathcal{S}_1^{n-1}$ with $\langle v, w\rangle = 0$. Then $\exists a, b, c \in \mathbb{R}$, so that the cross-entropy $\mathrm{H}\,(\text{Softmax}\,(w)\,|\text{Softmax}\,(SLERP\,(v,w,t))) = a \cdot \cos\left(\frac{\pi}{2}x + b\right) + c$ for $0 \leq t \leq 1$.

$$\theta = \angle(v, w) \quad \lambda_1(t) = \frac{\sin((1-t)\cdot\theta)}{\sin(\theta)} \quad \lambda_2(t) = \frac{\sin(t\cdot\theta)}{\sin(\theta)}$$
$$SLERP(v, w, t) = \lambda_1(t)\cdot v + \lambda_2(t)\cdot w \qquad 0 \leq t \leq 1$$

Proof: Let us abbreviate Softmax with Sm. The statement is equivalent to showing that $\exists \hat{a}, \hat{b} \in \mathbb{R}$, so that

$$\frac{\mathrm{dH}\left(Sm(w)\,|Sm\left(SLERP(v, w, t)\right)\right)}{\mathrm{d}t} = \hat{a}\cdot\cos\left(\frac{\pi}{2}x + \hat{b}\right), \qquad 0 \leq t \leq 1$$

Parametrizing the vector v in dependence of the time variable t:

$$v(t) = SLERP(v, w, t) = \left[v_1(t)\, v_2(t) \ldots v_n(t)\right]^T$$

To calculate $\frac{\mathrm{d}H\left(Sm(w)\,|Sm\left(v(t)\right)\right)}{\mathrm{d}t}$, we need to make use of several chain rule operations in the following scheme:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{\mathrm{d}H}{\mathrm{d}Sm(v_1)}\cdot\frac{\mathrm{d}Sm(v_1)}{\mathrm{d}v_1}\cdot\frac{\mathrm{d}v_1}{\mathrm{d}t} + \cdots + \frac{\mathrm{d}H}{\mathrm{d}Sm(v_n)}\cdot\frac{\mathrm{d}Sm(v_n)}{\mathrm{d}v_n}\cdot\frac{\mathrm{d}v_n}{\mathrm{d}t} \tag{5.4}$$

$\frac{\mathrm{d}H}{\mathrm{d}Sm(v_i)}\cdot\frac{\mathrm{d}Sm(v_i)}{\mathrm{d}v_i}$ can be obtained by differentiating Softmax with respect to one component. Applying the chain rule to each dimension i, each Softmax difference (alike the formulas in 5.2) gets scaled by the derivate of the component v_i with respect to the time t. We then finally obtained:

$$\frac{\mathrm{d}H\left(Sm(w)\,|Sm\left(v(t)\right)\right)}{\mathrm{d}t} = \sum_{i=1}^{n}\left(Sm(w)_i - Sm(v)_i\right)\cdot\frac{\mathrm{d}v_i(t)}{\mathrm{d}t} \tag{5.5}$$

Differentiation of SLERP:

$$\frac{\mathrm{d}SLERP(v, w, t)}{\mathrm{d}t} = \frac{\mathrm{d}\lambda_1(t)\cdot v}{\mathrm{d}t} + \frac{\mathrm{d}\lambda_2(t)\cdot w}{\mathrm{d}t} = \frac{\mathrm{d}\frac{\sin((1-t)\cdot\theta)}{\sin(\theta)}v}{\mathrm{d}t} + \frac{\mathrm{d}\frac{\sin(t\cdot\theta)}{sin(\theta)}w}{\mathrm{d}t} \tag{5.6}$$

Differentiating both terms yields:

$$\frac{\mathrm{d}\frac{\sin((1-t)\cdot\theta)}{\sin(\theta)}v}{\mathrm{d}t} = \frac{-\theta\cos\left((1-t)\,\theta\right)}{\sin(\theta)}v \tag{5.7}$$

$$\frac{\mathrm{d}\frac{\sin(t\cdot\theta)}{sin(\theta)}w}{\mathrm{d}t} = \frac{\theta\cos(t\theta)}{\sin(\theta)}w \tag{5.8}$$

We obtain in combination:

$$\frac{\mathrm{d}SLERP\left(v,w,t\right)}{\mathrm{d}t}=\frac{\theta}{\sin\left(\theta\right)}\left(\cos\left(t\theta\right)w-\cos\left(\left(1-t\right)\theta\right)v\right) \tag{5.9}$$

Using $\langle v,w\rangle=0\rightarrow\theta=\angle\left(v,w\right)=\frac{\pi}{2}$, as well as $\cos\left(t+\frac{\pi}{2}\right)=\sin\left(t\right)$

$$\frac{\mathrm{d}SLERP\left(v,w,t\right)}{\mathrm{d}t}=\frac{\pi}{2}\left(\cos\left(t\frac{\pi}{2}\right)w-\cos\left(\left(1-t\right)\frac{\pi}{2}\right)v\right)=\frac{\pi}{2}\left(\cos\left(t\frac{\pi}{2}\right)w-\sin\left(t\frac{\pi}{2}\right)v\right) \tag{5.10}$$

That means for one dimension i, it holds true that:

$$\frac{\mathrm{d}SLERP\left(v,w,t\right)_i}{\mathrm{d}t}=\frac{\pi}{2}\left(\cos\left(t\frac{\pi}{2}\right)w_i-\sin\left(t\frac{\pi}{2}\right)v_i\right) \tag{5.11}$$

Starting to transform that expression into one single cosine function. We need to find an amplitude A and a phase ϕ, so that:

$$\frac{\pi}{2}\left(\cos\left(t\frac{\pi}{2}\right)w_i-\sin\left(t\frac{\pi}{2}\right)v_i\right)=A\cdot\cos\left(t\frac{\pi}{2}+\phi\right) \tag{5.12}$$

For terms of the form $\alpha\cos\left(\lambda t\right)+\beta\sin\left(\lambda t\right)$, the amplitude A is $\sqrt{\alpha^2+\beta^2}$ and the phase ϕ is $\arctan\left(\frac{\beta}{\alpha}\right)$. In our case:

$$\alpha=\frac{\pi}{2}w_i\quad\beta=\frac{-\pi}{2}v_i,\quad A=\frac{\pi}{2}\sqrt{w_i^2+v_i^2}\quad\phi=\arctan\left(\frac{-v_i}{w_i}\right) \tag{5.13}$$

We end up with:

$$\frac{\pi}{2}\left(\cos\left(t\frac{\pi}{2}\right)w_i-\sin\left(t\frac{\pi}{2}\right)v_i\right)=\frac{\pi}{2}\sqrt{w_i^2+v_i^2}\cos\left(t\frac{\pi}{2}+\arctan\left(-\frac{v}{w}\right)\right) \tag{5.14}$$

$$0\leq\cos\left(\arctan\left(-\frac{v_i}{w_i}\right)\right)=\frac{1}{\sqrt{\left(\frac{v_i}{w_i}\right)^2+1}}\leq\frac{\pi}{2} \tag{5.15}$$

Combining all individual cosine terms in the summation formula, we obtain:

$$\frac{\mathrm{d}H\left(Sm\left(w\right)|Sm\left(v\left(t\right)\right)\right)}{\mathrm{d}t}=\sum_{i=1}^{n}\left(Sm\left(w\right)_i-Sm\left(v\right)_i\right)\cdot\frac{\pi}{2}\sqrt{w_i^2+v_i^2}\cos\left(t\frac{\pi}{2}+\arctan\left(-\frac{v}{w}\right)\right) \tag{5.16}$$

The sum of cosine functions is again a cosine function with the same frequency if all cosine terms have the same frequency, which is the case with $\theta=\frac{\pi}{2}$:

$$\exists\hat{a},\hat{b}\in\mathbb{R}\quad\frac{\mathrm{d}H\left(Sm\left(w\right)|Sm\left(SLERP\left(v,w,t\right)\right)\right)}{\mathrm{d}t}=\hat{a}\cdot\cos\left(\frac{\pi}{2}x+\hat{b}\right),\qquad0\leq t\leq1 \tag{5.17}$$

That concludes the proof. The same proof can be repeated for the statement $H\left(\text{Softmax}\left(v\right)|\text{Softmax}\left(SLERP\left(v,w,t\right)\right)\right)$ by swapping each v_i and w_i.

Corollary: Let $v, w \in \mathcal{S}_1^{n-1} | \langle v, w \rangle = 0$. Then $\exists a, b, c \in \mathbb{R}$, so that the KL divergence $D_{KL} = \left(\text{Softmax}\left(w\right)||\text{Softmax}\left(SLERP\left(v,w,t\right)\right)\right) = a \cdot \cos\left(\frac{\pi}{2}x + b\right) + c$ for $0 \leq t \leq 1$.

Proof: Use the property of the KL divergence $D_{KL}(p,q) = H(p,q) - H(p)$ to shift the cosine function by the fixed entropy of $H\left(\text{Softmax}\left(w\right)\right)$:

$$D_{KL}\left(\text{Softmax}\left(v\right)||\text{Softmax}\left(SLERP\left(v,w,t\right)\right)\right) \tag{5.18}$$

$$= H\left(\text{Softmax}\left(v\right)|\text{Softmax}\left(SLERP\left(v,w,t\right)\right)\right) - H\left(\text{Softmax}\left(w\right)\right) \tag{5.19}$$

So the corollary is proven.

That comes with the benefit that $D_{KL}\left(\text{Softmax}\left(v\right)||\text{Softmax}\left(SLERP\left(v,w,0\right)\right)\right) = 0$. The cosine course of the curve also applies to the Jensen Divergence $JSD\left(p,q\right) = \frac{1}{2}D_{KL}\left(p,m\right) + \frac{1}{2}D_{KL}\left(q,m\right)$, with $m = \frac{p+q}{2}$, which simply adds up two KL divergence terms or two cosine functions of same frequency.

The main insight is that optimizing the cross-entropy in the self-attention layer involves geometrically interpolating their respective vector representatives in Euclidean geometry in a direct reparametrization from the angle $\theta = \angle\left(v,w\right) = \frac{\pi}{2}$ because the frequency of the cosine function is $\frac{\pi}{2}$.

Models that try to learn certain target probability distributions are thus required to rotate the embedding vectors on the hypersphere to meet the target in the space of discrete probabilities, and embedding vectors that rotational and angular information on the hypersphere would also carry probabilistic information, which is then later used to build mixture distributions. The entropy of Softmax-transformed random vectors is almost always the same for random vectors close to that of a balanced distribution, and their entropy is very close to $\log\left(n\right)$, meaning all random vectors on a hypersphere encode almost the same level of information.

5.3 Statistics of Log-Normal Variables

5.3.1 Distribution of Ratios

The exact distribution of the probabilities p_i after applying the Softmax function is indeed complicated due to the normalization step. However, the ratio of probability distributions $\frac{p_i}{p_j}$ remains unaffected, as $\frac{p_i}{p_j} = \frac{\exp(x_i)}{\exp(x_j)} = \exp(x_i - x_j)$. Hence, the relational information is determined solely by the exponentiated distance between the vectors.

If a random variable is normally distributed, then its exponentiation is log-normally distributed, for which we already specified the following formulas for the first two statistical moments:

$$X \in \mathcal{N}(\mu, \sigma^2), \quad \mathbb{E}[\exp(X)] = \exp\left(\frac{\mu + \sigma^2}{2}\right) \quad \text{Var}(\exp(X)) = \left(\exp(\sigma^2) - 1\right) \cdot \exp\left(2\mu + \sigma^2\right)$$

If all x_i are distributed according to $\mathcal{N}(\mu_i, \sigma_i^2)$ with $\text{Cov}[x_i, x_j] = \sigma_{ij}$, then $x_i - x_j \in \mathcal{N}\left(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}\right)$, so that:

$$\mathbb{E}[\frac{p_i}{p_j}] = \exp\left(\frac{\mu_i - \mu_j + \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}}{2}\right) \tag{5.20}$$

$$\text{Var}\left(\frac{p_i}{p_j}\right) = \left(\exp\left(\sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}\right) - 1\right) \cdot \exp\left(2\left(\mu_i - \mu_j\right) + \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}\right) \tag{5.21}$$

For the classical Xavier initialization $v \sim \mathcal{N}(0, \frac{1}{n})$, we already derived that a vector v lies in expectation on a hypersphere with radius 1 - multiplication with a length parameter l sets them on the corresponding hypersphere with radius l. Thus we choose $\mu_i = \mu_j = 0$, $\sigma_i^2 = \sigma_j^2 = l \cdot \frac{1}{n}$, and $\sigma_{ij} = 0$. Then:

$$\mathbb{E}[\frac{p_i}{p_j}] = \exp\left(\frac{l}{n}\right) \quad \text{Var}\left(\frac{p_i}{p_j}\right) = \left(\exp\left(\frac{2l}{n}\right) - 1\right) \cdot \exp\left(\frac{2l}{n}\right)$$

A Taylor approximation of the exponential term of order 2 gives us $\exp\left(\frac{l}{n}\right) \approx 1 + \frac{l}{n} + \frac{l^2}{2n^2}$ and $\exp\left(\frac{2l}{n}\right) \approx 1 + \frac{2l}{n} + \frac{2l^2}{n^2}$, so that:

$$\mathbb{E}[\frac{p_i}{p_j}] \approx 1 + \frac{l}{n} + \frac{l^2}{2n^2}$$

$$\text{Var}\left(\frac{p_i}{p_j}\right) \approx \left(\frac{2l}{n} + \frac{2l^2}{n^2}\right) \cdot \left(1 + \frac{2l}{n} + \frac{2l^2}{n^2}\right) \approx \frac{2l}{n} + \frac{6l^2}{n^2} + \dots$$

The takeaway point is that for $l = 1$, the probability ratio $\frac{p_i}{p_j}$ between a_i and a_j is expected to hover around $1 + \frac{1}{n}$, with a variance of $\mathcal{O}\left(\frac{1}{n}\right)$ and a standard deviation of $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. The oscillation is not negligible and allows the model to manage fractional relationship information $\frac{p_i}{p_j}$ more sensibly. For $l < 1$, this bound becomes even tighter, and we expect very balanced distributions with high entropy as $\mathbb{E}[\frac{p_i}{p_j}]$ approaches 1. For $l > 1$, the probability ratio gets larger and oscillates more due to the higher-order Taylor terms.

If we look back at $\mathbb{E}[\frac{p_i}{p_j}] = \exp\left(\frac{\mu_i - \mu_j + \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}}{2}\right)$, the only chance for the model to distinctively separate the attention probability scores is to either shift the mean difference $\mu_i - \mu_j$ or, more decisively, alter the individual variance terms to increase $\sigma_i^2 + \sigma_j^2$ and outweigh the Pearson correlation ρ in $\sigma_{ij} = \frac{\rho}{\sigma_i \sigma_j}$. However, as our calculations demonstrate, increasing $\frac{p_i}{p_j}$ takes much effort because the expression is relatively rigid for $l \leq 1$.

5.3.2 Non-Linearity of exponentiation

After analyzing the value domain of the resulting probability distributions, we now examine the linear combinations of raw attention scores $z = \sum_{i=1}^{n} \lambda_i \cdot v_i$, where $v_i = \begin{bmatrix} a_1^{(i)} & \dots & a_n^{(i)} \end{bmatrix}^T = \begin{bmatrix} \exp\left(x_1^{(i)}\right) & \dots & \exp\left(x_n^{(i)}\right) \end{bmatrix}^T$ and $\sum_{i=1}^{n} \lambda_i = 1$. So, how non-linear is the exponentiation in our setting? A Taylor expansion of order 2 gives us ($v_i \cdot v_i = v_i^2$ means the element-weise Hadamard product $v_i \circ v_i$):

$$\exp\left(\lambda_i \cdot v_i\right) \approx 1 + \lambda_i v_i + \frac{\lambda_i^2 v_i^2}{2} = \lambda_i \cdot \left(1 + v_i + \frac{v_i^2}{2}\right) + (1 - \lambda_i) + \frac{\left(\lambda_i^2 - \lambda_i\right) v_i^2}{2} \approx$$

$$\lambda_i \cdot \exp\left(v_i\right) + 1 - \lambda_i + \left(\lambda_i^2 - \lambda_i\right) \cdot \frac{v_i^2}{2}$$

Multiplication turns out to be problematic, which appears logical due to the asymmetry introduced by the exponentiation, but what about addition?

Multivariate Taylor expansion of order 2 also gives us:

$$exp\left(z\right) = exp\left(\sum_{i=1}^{n} \lambda_i \cdot v_i\right) \approx 1 + \sum_{i=1}^{n} \left(\lambda_i \cdot v_i\right) + \frac{1}{2} \sum_{i=1}^{n} \left(\lambda_i^2 \cdot v_i^2\right) + \sum_{1 \leq i < j \leq n} \left(\lambda_i \cdot \lambda_j \cdot v_i \cdot v_j\right) \approx$$

Using that $\exp(\lambda_i \cdot v_i) \approx 1 + \lambda_i v_i + \frac{\lambda_i^2 v_i^2}{2}$:

$$\approx (n-1) + \sum_{i=1}^{n} \exp(\lambda_i v_i) + \sum_{1 \leq i < j \leq n} (\lambda_i \cdot \lambda_j \cdot v_i \cdot v_j)$$

We can see that the linear combinations get shifted by $n-1$ and distorted by a noise term, shaped by the covariant statistics of v_i and v_j, multiplied by the contribution of the product $\lambda_i \cdot \lambda_j$.

Let $v_i, v_j \in \mathcal{N}\left(0, diag\{\sigma_{i1}^2, \ldots, \sigma_{in}^2\}\right), \mathcal{N}\left(0, diag\{\sigma_{j1}^2, \ldots, \sigma_{jn}^2\right)\}, i.i.d.$ Applying the $\tilde{\chi}^2$-distribution for the square of two different normal variable yields:

$$\mathbb{E}[v_i v_j] = \begin{bmatrix} 0 & \ldots & 0 \end{bmatrix}^T \qquad \text{Var}(v_i v_j) = \begin{bmatrix} \sigma_{i1}^2 \sigma_{j1}^2 & \ldots & \sigma_{in}^2 \sigma_{jn}^2 \end{bmatrix}^T$$

Assuming that $\sigma_{ij}^2 \approx \frac{l}{n}$ as in the last subsection, $\text{Var}(v_i \cdot v_j) \approx \begin{bmatrix} \sigma_{i1}^2 \sigma_{j1}^2 & \ldots & \sigma_{in}^2 \sigma_{jn}^2 \end{bmatrix}^T \approx \begin{bmatrix} \frac{l^2}{n^2} & \ldots & \frac{l^2}{n^2} \end{bmatrix}^T$ and we can conclude:

$$\mathbb{E}[\cdot \sum_{1 \leq i < j \leq n} (\lambda_i \cdot \lambda_j \cdot v_i \cdot v_j)] = 0$$

Utilizing Jensen's inequality $\sum_{1 \leq i < j \leq n} (\lambda_i \lambda_j)^2 \leq \sum_{1 \leq i < j \leq n} \left(\frac{1}{n^2}\right)^2$:

$$\text{Var}\left(\sum_{1 \leq i < j \leq n} (\lambda_i \cdot \lambda_j \cdot v_i \cdot v_j)\right) \leq \text{Var}\left(\sum_{1 \leq i < j \leq n} \frac{1}{n} \cdot \frac{1}{n} (v_i \cdot v_j)\right) = \sum_{1 \leq i < j \leq n} \text{Var}\left(\frac{1}{n^2} v_i \cdot v_j\right)$$

$$\sum_{1 \leq i < j \leq n} \frac{1}{n^4} \text{Var}(v_i \cdot v_j) \approx \left(n^2 - n\right) \frac{1}{n^4} \cdot \frac{2l^2}{n^2} \in \mathcal{O}\left(\frac{l^2}{n^4}\right)$$

Thus, can we say that it is to some extent linear after the affine shift of $n-1$ in terms of contributions with respect to the new linear combination of exponentiated vectors $\exp(\lambda_1 v_i), \ldots, \exp(\lambda_n v_n)$ for $l \leq 1$, meaning linear combinations translate in an approximated direct relationship.

This technical derivation with the Taylor approximations all aimed to show that if vectors are located on a sphere with radius $l \leq 1$, linear combinations of vectors translate well-behaved into their exponentiated counterparts, even tighter for decreasing l. So, if a Transformer model passes linear combinations of vectors $\sum_{i=1}^{n} \lambda_i \cdot v_i$ or their respective distributions into an attention block, then it still recognizes those contributions in terms of $\sum_{i=1}^{n} \exp(\lambda_i v_i)$, disturbed by a negligible little noise term.

5.4 Summary Statistics and Geometry of Softmax in Attention

To summarize the findings in this chapter, the SoftMax operation involves two distinct processes: exponentiation and normalization. Understanding the distribution of attention scores necessitates examining log-likelihood distributions. Boltzmann-distributed variables enable the analysis of the ratios of raw attention scores, revealing that the distribution remains relatively flat compared to the magnitude of the Euclidean distance of the inputs to the SoftMax operation. Therefore, the resulting normalized attention scores exhibit a relatively balanced distribution. This flatness is influenced by the statistics of the input distributions, which we simulated using a hypersphere with a fixed radius l. The connection between the linearity of discrete probability distributions and vector combinations is particularly insightful, showing affine linearity not for multiplication but for addition under length-contracting hyperspheres with radius $l \leq 1$. We also explored how geometry in Euclidean space relates to that of probability spaces, which challenges might arise during that comparison, and how information theory can encounter those. We also explored spherical linear combinations through the lens of SLERP interpolation, demonstrating its relation to the cross-entropy of SoftMax-transformed discrete probability vectors. This provides a geometric perspective on what learning target probability distributions entails in the familiar Euclidean space, showing the direct dependence between the embedding cosine similarity or angle to the probability measure distance.

6 Attention as Bilinear Form with Randomized and Learned Weight Matrices

6.1 Attention Matching as Bilinear Form

Since we have thoroughly discussed the processing of linear combinations during the Softmax function, we now aim to draw conclusions about the vectors' statistics concerning the attention weights' parameters. By examining the vector statistics, we can better understand the model's capabilities.

6.1.1 Description Attention Matching

The query and key matrices are obtained by multiplying the query weight matrix $W_q \in \mathbb{R}^{d \times d_q}$ and the key weight matrix $W_k \in \mathbb{R}^{d \times d_k}$ with the embedded sequences $X \in \mathbb{R}^{n \times d}$, resulting in $Q = XW_q$ and $K = XW_k$. The final attention scores are obtained after applying the row-wise Softmax to $\frac{QK^T}{\sqrt{d_k}}$. Alternatively, attention can also be expressed for two sentence $\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$ and $\begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$, $x_i, y_i \in \mathbb{R}^d$, with $\langle x, y \rangle := x^T y$:

$$A = \frac{1}{\sqrt{d_k}} \begin{bmatrix} \langle W_q x_1, W_k x_1 \rangle & \langle W_q x_1, W_k x_2 \rangle & \dots & \langle W_q x_1, W_k x_n \rangle \\ \langle W_q x_2, W_k x_1 \rangle & \langle W_q x_2, W_k x_2 \rangle & \dots & \langle W_q x_2, W_k x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle W_q x_n, W_k x_1 \rangle & \langle W_q x_n, W_k x_2 \rangle & \dots & \langle W_q x_n, W_k x_n \rangle \end{bmatrix}$$

We now focus just again on one row i:

$$A_i = \frac{1}{\sqrt{d_k}} \begin{bmatrix} \langle W_q x_i, W_k x_1 \rangle & \langle W_q x_i, W_k x_2 \rangle & \dots & \langle W_q x_i, W_k x_n \rangle \end{bmatrix} \tag{6.1}$$

The dot product defines an asymmetric bilinear form $B(x, y) = \langle W_q x, W_k y \rangle = x^T W_q^T W_k y$, which is profoundly influenced by the pair-wise alignment $W_q^T W_k$ of the column vector space of W_k and the row vector space of W_q.

6.1.2 Attention: An asymmetric bilinear form

In this subsection, we break down the expression: $\langle W_q x, W_k y \rangle = x^T W_q^T W_k y$, understanding how multifaceted the bilinear form is by setting different probability distributions for the embeddings. For conciseness, we will use $A = W_q$ and $B = W_k$ for the rest of this section. And for the sake of simplicity, we set $d_q = d_k = d$, making our weight matrices quadratic with shape $d \times d$, to which we will now refer as n.

We begin with randomized weight matrices with Xavier Initialization [21], as discussed in the Random Matrix Background Section 2.5, where each element is distributed according to $\mathcal{N}(0, \sigma^2)$ and contrast them with learned weight matrices. We cover the following two cases for two embedding vectors $x, y \in \mathbb{R}^n$ in the bilinear form $\langle Ax, By \rangle$:

1. **Multivariate Gaussians:**

$$\begin{pmatrix} x \in \mathbb{R}^n \\ y \in \mathbb{R}^n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix} \right) \text{ and } A, B \in \mathbb{R}^{n \times n}, C = A^T B$$

2. **Hyperspheres:**

$$\{(x, y) \in \mathcal{S}_{l_1}^{n-1} \times \mathcal{S}_{l_2}^{n-1} | \langle x, y \rangle = k \cdot l_1 \cdot l_2\}, \quad -1 \leq k \leq 1$$

For the latter, vector pairs are drawn with equal probability from the $n-1$ dimensional hypersphere $\mathcal{S}_l^{n-1} = \{x \in \mathbb{R}^n | \quad ||x||_2 = l\}$ of radius l.

6.2 Bilinear Form Statistics Case 1: Multivariate Gaussians

6.2.1 Expected value and Variance for Bilinear Forms with Gaussians

We start with the case where the unit vectors are drawn from dependent multivariate Gaussians. For the following proof, I verified and improved the calculations using the OpenAI tool ChatGPT4.

Statement:

Let $\begin{pmatrix} x \in \mathbb{R}^n \\ y \in \mathbb{R}^n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}\right)$ and $A, B \in \mathbb{R}^{n \times n}$, with $C = A^T B$

Then it holds:

$$\mathbb{E}[\langle Ax, By \rangle] = \mu_X^T C \mu_Y + \operatorname{Tr}(C\Sigma_{XY}) \tag{6.2}$$

And for $\Sigma_{XY} = \Sigma_{YX} = 0$:

$$\operatorname{Var}(\langle Ax, By \rangle) = \operatorname{Tr}\left(\Sigma_X C \Sigma_Y C^T\right) + \mu_Y^T C^T \Sigma_X C \mu_Y + \mu_X^T C \Sigma_Y C^T \mu_X \tag{6.3}$$

Proof of the first property 6.2:

$$\langle Ax, By \rangle = x^T A^T B y = x^T C y + \operatorname{Tr}(C\Sigma_{XY}) \tag{6.4}$$

Since the trace is a scalar value such as a quadratic form $\langle x, y \rangle_M = x^T M y$, and using the trace's cyclic property $\operatorname{Tr}(ABC) = \operatorname{Tr}(BCA)$, we get:

$$\mathbb{E}[\langle Ax, By \rangle] = \mathbb{E}[\operatorname{Tr}\left(x^T C y\right)] = \mathbb{E}[\operatorname{Tr}\left(C y x^T\right)] \tag{6.5}$$

Since the trace sums up how much each row vector C_i of C scales the column vector i from the outer vector product yx^T; by rearranging the summated terms, we can move the linear map C out of the linear trace operator.

$$\mathbb{E}[\operatorname{Tr}\left(C y x^T\right)] = \operatorname{Tr}\left(C \mathbb{E}[y x^T]\right) \tag{6.6}$$

Using the property of outer products from Gaussian variables $\mathbb{E}[yx^T] = \mu_Y \mu_X^T + \Sigma_{XY}$

$$\operatorname{Tr}\left(C\mathbb{E}[yx^T]\right) = \operatorname{Tr}\left(C\left(\mu_Y \mu_X^T + \Sigma_{XY}\right)\right) \tag{6.7}$$

Linearity and cyclic property of the trace operator:

$$\operatorname{Tr}\left(C\left(\mu_Y \mu_X^T + \Sigma_{XY}\right)\right) = \operatorname{Tr}\left(C\mu_Y \mu_X^T\right) + \operatorname{Tr}\left(C\Sigma_{XY}\right) = \operatorname{Tr}\left(\mu_X^T C \mu_Y\right) + \operatorname{Tr}\left(C\Sigma_{XY}\right) \tag{6.8}$$

Since the trace is a scalar number in the first term:

$$\mathbb{E}[\langle Ax, By\rangle] = \cdots = \operatorname{Tr}\left(\mu_X^T C \mu_Y\right) + \operatorname{Tr}\left(C\Sigma_{XY}\right) = \mu_X^T C \mu_Y + \operatorname{Tr}\left(C\Sigma_{XY}\right) \tag{6.9}$$

That concludes the proof. A proof of the second property 6.3 (predominantly GPT4-assisted) can be found in Appendix B.1.

Noteworthy, if both Gaussians are independent and have zero mean, then $\mathbb{E}[\langle Ax, By\rangle] = 0$ and $\operatorname{Var}\left(\langle Ax, By\rangle\right) = \operatorname{Tr}\left(\Sigma_X A^T B \Sigma_Y B^T A\right) = \operatorname{Tr}\left(\Sigma_X C \Sigma_Y C^T\right)$. This expression is significant as it expresses how the alignment matrix C interacts with both Gaussian Covariance matrices Σ_X and Σ_Y. But why call C a base alignment matrix?

This terminology stems from the fact that, since the quadratic matrices A and B are diagonalizable (applicable to both random and learned matrices), we can express them in terms of their row and column vector space, which form a base: $A^T = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix}^T$ and $B = \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{bmatrix}$. Thus the matrix element $c_{ij} = C_{ij}$ encodes the alignment, cosine similarity, or respective angle between a_i and b_j, scaled by their lengths $||a_i||_2$ and $||b_j||_2$. Consequently, the trace adds up all alignment scores by:

$$\operatorname{Tr}\left(C\right) = \sum_{i=1}^{n} c_{ii} = \operatorname{Tr}\left(A^T B\right) = \sum_{i=1}^{n} \langle a_i, b_i\rangle = \sum_{i=1}^{n} a_i^T b_i \tag{6.10}$$

If A and B were orthonormal $\operatorname{Tr}\left(A^T B\right) \approx n$ implies complete alignment due to $\langle a_i, b_i\rangle \approx 1$ and effectively $A^T B \approx \mathrm{Id}$. Conversely, $\operatorname{Tr}\left(A^T B\right) \approx 0$ indicates that the vectors are nearly orthogonal pairwise, comparable to the observation in formula 5.3.

Much revolves around how models effectively adjust $\operatorname{Tr}\left(C\right) = \operatorname{Tr}\left(A^T B\right)$ and what this means for scaling capabilities in the subsequent attention module. The same property holds for non-orthogonal matrices if the base length vec-

tors $||a_i||_2$ and $||b_i||_2$ are omitted, corresponding to the squared singular values of A and B. Furthermore, it implies that models alternating the singular value spectra of A and B effectively scale $\text{Tr}\left(A^T B\right)$ in this regard.

Additionally, in case Σ_X and Σ_Y are nearly diagonal, the variance of the scaling $\langle x, y \rangle$ would predominantly depend on

$$\text{Tr}\left(C^T C\right) = \sum_{i,j=1}^{n,n} c_{ij} c_{ij} = \sum_{i,j=1}^{n,n} \langle a_i, b_j \rangle^2 = ||C||_F^2 \tag{6.11}$$

, corresponding to the squared Frobenius norm $||C||_F^2$of the base alignment matrix C.

Next, we examine how these expressions change when both matrices A and B are random and how the singular value spectra of both affect the scaling capabilities on base alignments.

6.2.2 Random Matrix Distribution of Singular Values

Now, we move to the distribution of singular values in a random matrix. Concerning distribution of singular values is or a random matrix $A \in \mathbb{R}^{n \times n}$, with each element i.d.d distributed via $\mathcal{N}\left(0, \sigma^2\right)$ supported by my numerical evaluations and the Random Matrix Theory sources [38, 24], yields the formulas:

Statement: Let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be the descendingly ordered set of singular values from the random matrix $A \in \mathbb{R}^{n \times n}$ ($A_{ij}, i.i.d. \sim \mathcal{N}\left(0, \sigma^2\right)$), then, converging for $n \to \infty$, it holds:

$$\mathbb{E}[\max\left(\sigma_i^2\right)] = \mathbb{E}[\sigma_1^2] \approx 4n\sigma^2 \tag{6.12}$$

Moreover, we have that:

$$\mathbb{E}[\sum_{i=1}^{n} \sigma_i^2] = n^2\sigma^2 \qquad \text{Var}\left(\sum_{i=1}^{n} \sigma_i^2\right) = 2n^2\sigma^4 \tag{6.13}$$

Proof:

The probability density function in the Random Matrix Theory Background section 2.5 yields the approximated maximum squared singular value.

Concerning the latter two formulas: $\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2] = n\sigma^2$ and $\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2\right) = 2\sigma^4$ can be obtained by using the property that the sum of squared singular values is equal to the trace of A^TA.

$$\sum_{i=1}^{n}\sigma_i^2 = \mathrm{Tr}\left(A^TA\right) \tag{6.14}$$

One matrix element a_{ij} of A^TA is distributed according to the $\tilde{\chi}^2\,(n)$-distribution of a squared normal variable, which has a mean of σ^2 and a variance of $2\sigma^4$.

$$\mathbb{E}[\mathrm{Tr}\left(A^TA\right)] = \mathbb{E}[\sum_{i,j=1,1}^{n,n} a_{ij}] = \sum_{i,j=1,1}^{n,n}\mathbb{E}[a_{ij}] = \sum_{i,j=1,1}^{n,n}\sigma^2 = n^2\sigma^2 \tag{6.15}$$

$$\mathrm{Var}\left(\mathrm{Tr}\left(A^TA\right)\right) = \mathrm{Var}\left(\sum_{i,j=1,1}^{n,n} a_{ij}\right) = \sum_{i,j=1,1}^{n,n}\mathrm{Var}\left(a_{ij}\right) = \sum_{i,j=1,1}^{n,n} 2\sigma^4 = 2n^2\sigma^4 \tag{6.16}$$

That concludes the proof.

For Xavier initialization ($\mathcal{N}\left(0, \frac{1}{n}\right)$), we get $\mathbb{E}[\sigma_1^2] \approx 4$, $\mathbb{E}[\sum_{i=1}^{n}\sigma_i^2] = n$, as well as $\mathrm{Var}\left(\sum_{i=1}^{n}\sigma_i^2\right) = 2$.

6.2.3 Scaling Behavior of the Random Matrix Product

First, we must discuss the product of two random matrices A^TB, determining the bilinear form $\langle Ax, By\rangle$.

Statement: For two randomized matrices $A, B \in \mathbb{R}^{n\times n}$ ($A_{ij}, B_{ij}, i.i.d. \sim \mathcal{N}\left(0, \sigma^2\right)$), we can obtain:

$$\mathbb{E}[\mathrm{Tr}\left(A^TB\right)] = 0 \qquad \mathrm{Var}\left(\mathrm{Tr}\left(A^TB\right)\right) = n^2\sigma^4 \tag{6.17}$$

Proof $\mathbb{E}[\mathrm{Tr}\left(A^TB\right)] = 0$:

$$\mathbb{E}[\mathrm{Tr}\left(A^TB\right)] = \mathbb{E}[\sum_{i,j=1,1}^{n,n} a_{ij}b_{ij}] = \sum_{i,j=1,1}^{n,n}\mathbb{E}[a_{ij}b_{ij}] = \sum_{i,j=1,1}^{n,n}\mathbb{E}[a_{ij}]\mathbb{E}[b_{ij}] = 0 \tag{6.18}$$

Proof $\mathrm{Var}\left(\mathrm{Tr}\left(A^TB\right)\right) = n^2\sigma^4$:

$$\text{Var}\left(\text{Tr}\left(A^T B\right)\right) = \mathbb{E}[\text{Tr}\left(A^T B\right)^2] - \left(\mathbb{E}[\text{Tr}\left(A^T B\right)]\right)^2 = \mathbb{E}[\text{Tr}\left(A^T B\right)^2] \tag{6.19}$$

Using linearity of the expected value and splitting the term up into all different index pairs (suggestion and notation from GPT4):

$$\mathbb{E}[\text{Tr}\left(A^T B\right)^2] = \mathbb{E}[\left(\sum_{i,j=1,1}^{n,n} a_{ij} b_{ij}\right)^2] = \mathbb{E}[\sum_{i,j=1,1}^{n,n} a_{ij}^2 b_{ij}^2] + \mathbb{E}[\sum_{(i,j)\neq(k,l)}^{n,n} a_{ij} a_{kl} b_{ij} b_{kl}] \tag{6.20}$$

The first term involves quadratic elements being independent, so chi-squared distributed:

$$\mathbb{E}[\sum_{i,j=1,1}^{n,n} a_{ij}^2 b_{ij}^2] = \sum_{i,j=1,1}^{n,n} \mathbb{E}[a_{ij}^2]\mathbb{E}[b_{ij}^2] = \sum_{i,j=1,1}^{n,n} \sigma^2\sigma^2 = n^2\sigma^4 \tag{6.21}$$

The second involves the multiplication of $i.i.d$ matrix elements, and $E[a_{ij}] = E[b_{ij}] = 0$ gives us finally:

$$\mathbb{E}[\sum_{(i,j)\neq(k,l)}^{n,n} a_{ij} a_{kl} b_{ij} b_{kl}] = \sum_{(i,j)\neq(k,l)}^{n,n} \mathbb{E}[a_{ij}]\mathbb{E}[a_{kl}]\mathbb{E}[b_{ij}]\mathbb{E}[b_{kl}] = 0 \tag{6.22}$$

Collecting all individual terms:

$$\text{Var}\left(\text{Tr}\left(A^T B\right)\right) = n^2\sigma^4 + 0 = n^2\sigma^4 \tag{6.23}$$

That concludes the second proof.

For Xavier initialization ($\mathcal{N}\left(0, \frac{1}{n}\right)$), we get $\mathbb{E}[\text{Tr}\left(A^T B\right)] = 0$ and $\text{Var}\left(\text{Tr}\left(A^T B\right)\right) = 1$, which is reminiscent of a standard normal distribution. That suggests that matrices with a product trace behaving according to a standard normal distribution can be considered unrelated, such as in a randomized system.

How does the randomness of the matrices affect expected value and variance in the derived equations 6.2 and 6.3? Unfortunately, we need to draw more conclusions about the marginal covariance matrices Σ_X and Σ_Y to solve that problem.

6.2.4 Upscaling effects for Self-Pairings and Dependent Pairings

Now we plug the same embedding vector into the bilinear form $\langle Ax, Bx\rangle$ with $C = A^T B$, We want to understand how embeddings attend to themselves, allowing the model to distinguish them from pair-wise different embeddings.

Statement: Let $A, B \in \mathbb{R}^{n\times n}$, with $C = A^T B$, and $x \in \mathbb{R}^n \sim \mathcal{N}(\mu_X, \Sigma_X)$, then:

$$\mathbb{E}[\langle Ax, Bx\rangle] = \mathrm{Tr}\,(C\Sigma_X) + \mu_X^T C \mu_X$$

Proof: This equation can be shown similarly to the earlier proof of the property 6.2 by using the property of the outer vector product $E[xx^T] = \mu_X \mu_X^T + \Sigma_X$ instead.

The expected value term is particularly interesting because it effectively expresses that the scaling of self-wise attention is raised by the alignment of Σ_X and the base alignment matrix $C = A^T B$. That means that the base systems $\{a_i\}$ and $\{b_i\}$ can differ and still exhibit upscaling effects. If Σ_X as a covariance matrix was diagonal-dominant, then both base system $\{a_i\}$ and $\{b_i\}$ would need to be mainly orthogonal to each other to increase $\mathrm{Tr}\left(A^T B \Sigma_X\right)$.

From the perspective of the entire embedding sequence with n different Gaussian's for a complete forward pass, we can highlight the model's capabilities in $A^T B$ by how strong they align with each respective Gaussian $\mathcal{N}(\mu_i, \Sigma_i)$.

The same conclusions can be made about dependent Gaussians with higher singular values in their cross-covariance matrix Σ_{XY} also causing an upshift of the expected value of the bilinear form. In an abstract sense, the singular value spectra should be, on average, reflected in attention patterns as higher output values from the bilinear form tie with higher attention scores. This means that covariant relationships, which we also measure with the determinant fractions concerning Mutual Information, significantly impact the structure of attention patterns. However, a distinction needs to be made since the scaling properties are shaped by the trace, not the determinant, making the interdependencies more subtle.

6.2.5 The chances and problems of the Singular Value Decomposition on Weight Matrices

We can observe similar behaviors when inspecting the singular value decomposition of $A = U_A D_A V_A^T$ and $B = U_B D_B V_B^T$ with $A_{ij}, B_{ij}, i.i.d. \sim \mathcal{N}(0, \sigma^2)$. An intuitive view of the matrix spectral properties is to consider how the model upscales vectors by aligning them to high singular values with the right singular value matrices V_A and V_B. Examining tr $(A^T B)$ more closely with the singular value decomposition, we obtain:

$$\text{Tr}\left(A^T B\right) = \text{Tr}\left(V_A D_A U_A^T U_B D_B V_B^T\right) = \text{Tr}\left(D_A U_A^T U_B D_B V_B^T V_A\right)$$

We can see that the individual alignment scores $\text{Tr}\left(V_B^T V_A\right)$ and $\text{Tr}\left(U_A^T U_B\right)$ contribute to the total trace term, but it is not possible to relate those individual trace terms to the global formula, which is incorporated in the trace operator itself.

Statement: Let $A = U_A D_A V_A^T$ and $B = U_B D_B V_B^T$ be randomized linear maps of shape $n \times n$ with each element distributed according to $i.i.d \sim \mathcal{N}(0, \sigma^2)$ and V_A and V_B their right singular value matrices. Then there exist left singular value matrices U_A and U_B so that $\mathbb{E}[\text{Tr}\left(A^T B\right)] = n^2\sigma^2$

Proof:

$$\text{Choose: } U_A \text{ and } U_B \text{ with } U_A^T U_B = \left(V_B^T V_A\right)^{-1} = V_A^T V_B \tag{6.24}$$

Then define $W = V_B^T V_A$:

$$\text{Tr}\left(A^T B\right) = \text{Tr}\left(V_A D_A U_A^T U_B D_B V_B^T\right) = \text{Tr}\left(D_A U_A^T U_B D_B V_B^T V_A\right) \tag{6.25}$$

$$= \text{Tr}\left(D_A V_A^T V_B D_B V_B^T V_A\right) = \text{Tr}\left(D_A W^T D_B W\right) \tag{6.26}$$

Let $\sigma_{a1}, \ldots, \sigma_{an}$ and $\sigma_{b1}, \ldots, \sigma_{bn}$ denote the singular values of A and B.

$$D_A W^T D_B W = \begin{bmatrix} | & | & & | \\ \sigma_{a1}\mathbf{w}_1 & \sigma_{a2}\mathbf{w}_2 & \cdots & \sigma_{an}\mathbf{w}_n \\ | & | & & | \end{bmatrix}^T \cdot \begin{bmatrix} | & | & & | \\ \sigma_{b1}\mathbf{w}_1 & \sigma_{b2}\mathbf{w}_2 & \cdots & \sigma_{bn}\mathbf{w}_n \\ | & | & & | \end{bmatrix} \tag{6.27}$$

Due to the row vectors w_i being orthonormal, we obtain with $\langle \sigma_{ai}\mathbf{w}_i, \sigma_{bi}\mathbf{w}_i \rangle = \sigma_{ai}\sigma_{bi}$:

$$\mathbb{E}[\mathrm{Tr}\left(D_A W^T D_B W\right)] = \mathbb{E}[\sum_{i=1}^{n} \langle \sigma_{ai}\mathbf{w}_i, \sigma_{bi}\mathbf{w}_i \rangle] = \sum_{i=1}^{n} \mathbb{E}[\sigma_{ai}\sigma_{bi}] = \sum_{i=1}^{n} \mathbb{E}[\sigma_{ai}]\mathbb{E}[\sigma_{bi}] \tag{6.28}$$

Using the characteristic 6.13 of random matrices $\mathbb{E}[\sum_{i=1}^{n} \sigma_i^2] = n^2\sigma^2 \rightarrow \mathbb{E}[\sigma_i^2] = n\sigma^2 \rightarrow \mathbb{E}[\sigma_i] = \sqrt{n}\sigma$ (unordered singular values, with $\mathbb{E}$ sampling over $\{\sigma_i\}$):

$$\mathbb{E}[\mathrm{Tr}\left(A^T B\right)] = \cdots = \sum_{i=1}^{n} \mathbb{E}[\sigma_{ai}]\mathbb{E}[\sigma_{bi}] = \sum_{i=1}^{n} \sqrt{n}\sigma\sqrt{n}\sigma = \sum_{i=1}^{n} n\sigma^2 = n^2\sigma^2 \tag{6.29}$$

That concludes the proof.

Applying the proof to arbitrary diagonalized bases U_A and U_B instead demonstrates the same result. Furthermore, for Xavier initialization ($\mathcal{N}\left(0, \frac{1}{n}\right)$), we would get $\mathbb{E}[\mathrm{Tr}\left(A^T B\right)] = n^2 \cdot \frac{1}{n} = n$, consistent with our discussion for completely aligned orthonormal matrices in the proof of statement 6.10.

The key point is that $\mathbb{E}[\mathrm{Tr}\left(A^T B\right)]$ is independent of $\mathrm{Tr}\left(V_B^T V_A\right)$ and $\mathrm{Tr}\left(U_A^T U_A\right)$. Regardless of the diagonalizable systems in which the embedding vectors are evaluated, the rotational difference $V_B^T V_A$ can be offset by the rotational difference $U_A^T U_B$. Therefore, these should only be considered together. While the singular value spectrum elucidates the scaling capabilities, the rotational matrices themselves are only insightful when coupled, which ties in with the idea of the division of degrees of freedom:

One possible way to think about a diagonalizable matrix $A \in \mathbb{R}^{n \times n}$ is as a joint probability distribution with n^2 degrees of freedom for each entry dimension or marginal distribution. Then the SVD $A = UDV^T$ divides the degrees of freedom among U, D, V, with D having n degrees of freedom and both U and V as orthonormal matrices with $\frac{n(n-1)}{2}$ degrees of freedom. Nonetheless, the singular value decomposition shifts those degrees of freedom at least into a more reasonable representation.

Therefore, the original assumption that the model learns a dual base through the right singular value matrix to the Gaussian base Σ_X is problematic. The model possesses the capability to alter the behavior of the bilinear form $\langle Ax, By \rangle$, observable through the "camera matrices" V_A and V_B from the diagonal spaces D_A and D_B, by manipulating the left singular matrices U_A and U_B.

If models change the singular value spectrum during training, orthonormal subspaces are created and proportionally upscaled in contrast to a randomized system. However, there might still be the issue that there could be left orthonormal singular value matrix subspaces that reverse or strengthen this effect, which warrants further scrutiny when studying the spectral properties of learned weight matrices.

Statement: Let $A = U_A D_A V_A^T$ and $B = U_B D_B V_B^T$ be random $n \times n$ matrices, with $A_{ij}, B_{ij}, i.i.d \sim \mathcal{N}(0, \sigma^2)$. If a model manipulates just the singular value spectrum so that $\hat{D_A} = diag\{\hat{\sigma_{a1}}, \dots, \hat{\sigma_{an}}\}$ and $\hat{D_B} = diag\{\hat{\sigma_{b1}}, \dots, \hat{\sigma_{bn}}\}$, then for $\hat{A}^T\hat{B} = \left(U_A\hat{D_A}V_A^T\right)^T\left(U_B\hat{D_B}V_B^T\right) = V_A\hat{D_A}U_A^T U_B\hat{D_B}V_B^T$, it holds that $\mathbb{E}[\mathrm{Tr}\left(\hat{A}^T\hat{B}\right)] = 0$ and $\mathrm{Var}\left(\mathrm{Tr}\left(\hat{A}^T\hat{B}\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\hat{\sigma}_{ai}^2\hat{\sigma}_{bi}^2$.

Proof: Define $W_V = V_B^T V_A$ and $W_U^T = \left(U_A^T U_B\right)^T$, so that (see 6.25):

$$\mathrm{Tr}\left(\hat{A}^T\hat{B}\right) = \mathrm{Tr}\left(V_A\hat{D_A}U_A^T U_B\hat{D_B}V_B^T\right) = \mathrm{Tr}\left(\hat{D_A}W_U^T\hat{D_B}W_V\right) \tag{6.30}$$

$$\hat{D_A}W_U^T\hat{D_B}W_V = \begin{bmatrix} | & | & & | \\ \hat{\sigma}_{a1}\mathbf{w}_{U1} & \hat{\sigma}_{a2}\mathbf{w}_{U2} & \cdots & \hat{\sigma}_{an}\mathbf{w}_{Un} \\ | & | & & | \end{bmatrix}^T \cdot \begin{bmatrix} | & | & & | \\ \hat{\sigma}_{b1}\mathbf{w}_{V1} & \hat{\sigma}_{b2}\mathbf{w}_{V2} & \cdots & \hat{\sigma}_{bn}\mathbf{w}_{Vn} \\ | & | & & | \end{bmatrix} \tag{6.31}$$

Since all possible orientations for the orthonormal matrices in V_B, V_A, U_A, and U_B are of equal probability, the property 5.3 applies to inner product of the respective column spaces $\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi} \rangle$:

$$\mathbb{E}[\mathrm{Tr}\left(\hat{A}^T\hat{B}\right)] = \mathbb{E}[\sum_{i=1}^{n}\hat{\sigma}_{ai}\hat{\sigma}_{bi}\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi}\rangle] = \sum_{i=1}^{n}\hat{\sigma}_{ai}\hat{\sigma}_{bi}\mathbb{E}[\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi}\rangle] = \sum_{i=1}^{n}\hat{\sigma}_{ai}\hat{\sigma}_{bi}0 = 0 \tag{6.32}$$

$$\mathrm{Var}\left(\mathrm{Tr}\left(\hat{A}^T\hat{B}\right)\right) = \mathrm{Var}\left(\sum_{i=1}^{n}\hat{\sigma}_{ai}\hat{\sigma}_{bi}\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi}\rangle\right) = \sum_{i=1}^{n}\hat{\sigma}_{ai}^2\hat{\sigma}_{bi}^2\mathrm{Var}\left(\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi}\rangle\right) \tag{6.33}$$

$$\sum_{i=1}^{n}\hat{\sigma_{ai}}^2\hat{\sigma_{bi}}^2\mathrm{Var}\left(\langle \mathbf{w}_{Ui}, \mathbf{w}_{Vi}\rangle\right) = \sum_{i=1}^{n}\hat{\sigma_{ai}}^2\hat{\sigma_{bi}}^2\frac{1}{n} = \frac{1}{n}\sum_{i=1}^{n}\hat{\sigma_{ai}}^2\hat{\sigma_{bi}}^2 \tag{6.34}$$

That concludes the proof.

The consequence is that, in case the left and right singular vector matrices behave like in a randomized system, alternating the singular value spectra translates directly into scaling effects of $\langle Ax, By\rangle$, as demonstrated in equation 6.2. More noteworthy, $\mathrm{Var}\left(\mathrm{Tr}\left(\hat{A}^T\hat{B}\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\hat{\sigma}_{ai}^2\hat{\sigma}_{bi}^2$ can also be understood as the squared dot product when considering the squared singular value

spectra $\hat{D_A} = diag\{\hat{\sigma^2_{a1}}, \dots, \hat{\sigma^2_{an}}\}$ and $\hat{D_B} = diag\{\hat{\sigma^2_{b1}}, \dots, \hat{\sigma^2_{bn}}\}$ as individual vectors. That allows the classification of singular value spectra in terms of magnitude $|\hat{D_A}|$ and $|\hat{D_B}|$ and an orientation on a hypersphere with $\mathcal{S}_1^{n-1}$, with the angle between two singular value spectra describing who they intertwine with each other concerning their alignment in $\text{Tr}\left(\hat{A}^T\hat{B}\right)$.

6.3 Bilinear Form Statistics Case 2: Hyperspheres

6.3.1 Hyperspheres and pre-trained angular information

Now we sample two vectors from $\{(x, y) \in \mathcal{S}_{l_1}^{n-1} \times \mathcal{S}_{l_2}^{n-1} | \langle x, y\rangle = k \cdot l_1 \cdot l_2\}, \quad -1 \leq k \leq 1$ with radius l_1 and l_2, respectively. This aims to explore how models can process the a priori angular or rotational information $\langle x, y\rangle$ of two embedding vectors x and y in the bilinear form Ax, By.

If we sample from this distribution, we would observe marginal distributions (where x_i denotes the i'th element of x):

$$\mathbb{E}[x_i] = \mathbb{E}[y_i] = 0 \qquad \text{Var}\left(x_i\right) = E[x_i^2] = \frac{l_1^2}{n} \qquad \text{Var}\left(y_i\right) = E[y_i^2] = \frac{l_2^2}{n} \tag{6.35}$$

As all orientations are equally probable, the i'th components contributes on average $\frac{1}{n}$-percent to the squared L^2-norm, explaining $E[x_i^2] = \frac{l_1^2}{n}$ and $E[x_i^2] = \frac{l_2^2}{n}$.

To understand the constraints on the individual terms x_i, inspecting that each hypersphere system has just $n-1$ degrees of freedom, we have:

$$2 \sum_{1 \leq i < j = n} \text{Cov}[x_i, x_j] = -\sum_{i=1}^{n} \text{Var}\left(x_i\right) = -\sum_{i=1}^{n} \frac{l_1^2}{n} = -l_1^2 \longrightarrow \text{Cov}[x_i, x_j] = -\frac{l_1^2}{n\left(n-1\right)} \tag{6.36}$$

Concerning the covariance of the vector components:

$$\mathbb{E}[xy] = kl_1l_2 \rightarrow \mathbb{E}[x_iy_i] = \frac{kl_1l_2}{n} \longrightarrow \text{Cov}[x_i, y_i] = E[x_iy_i] - E[x_i]E[y_i] = \frac{kl_1l_2}{n} \tag{6.37}$$

$$\forall i \neq j : \text{Cov}[x_i, y_j] = 0 \tag{6.38}$$

For different indices $i \neq j$, $\text{Cov}[x_i, y_j] = 0$, since the points are nearly independently distributed on their respective hyperspheres.

Collecting all terms together, the cross-covariance matrix Σ_{XY} becomes diagonal-dominant. Referring back to how the base alignment $A^T B$ relates to the cross-covariance matrices as shown in formula 6.2, we can deduce means that the bases $\{a_i\}$ and $\{b_i\}$ need to mainly orthogonal aligned to allow angular information to pass through.

However, this hypersphere simplification is unrealistic for real embedding distributions, as embeddings are pre-trained to yield specific outer vector products so that the singular value spectrum in the cross-covariance matrices becomes bent. Simply put, the eigenspace of the Gram matrix $X^T X$, which gives us only the angular information of the vector embeddings, does not straightforwardly relate to the eigenspace of the covariance matrix XX^T. The outer vector product xy^T offers additional information crucial for the model's functioning, we can't access that easily, consistent with the division of degrees of freedom in the singular value decomposition.

6.3.2 Processing of Pre-trained Angular Information

Just let us set $l_1 = l_2 = l = 1$ for the hyperspheres and consider $x \neq y$ for the following section. As we already covered the upscaling of $\langle Ax, Bx \rangle$, we now aim to analyze how attention scores for different vector pairings evolve. As we already derived in the proof of 6.2 for an arbitrary base alignment linear map C:

$$\mathbb{E}[x^T C y] = \mu_X^T C \mu_Y + \operatorname{Tr}(C \Sigma_{XY})$$

If the Gaussians $\mathcal{N}(\mu_X, \Sigma_X)$ and $\mathcal{N}(\mu_Y, \Sigma_Y)$ are dominated by k embedding vectors of unit length in form of a probabilistic linear combination $\mu_X = \sum_{i=1}^{k} \lambda_i v_i$ and $\mu_Y = \sum_{i=1}^{k} \tilde{\lambda}_i w_i$ with $\sum_{i=1}^{k} \lambda_i = 1$ and $\sum_{i=1}^{k} \tilde{\lambda}_i = 1$, we can constrain them onto hyperspheres to better study how positional information is processed. Thus, we sample from the following set:

$$\{((v_1, \ldots, v_k), (w_1, \ldots, w_k)) | \forall i, j : v_i, w_j \in \mathcal{S}_1^{n-1} \times \mathcal{S}_1^{n-1} | \langle v_i, w_j \rangle = k_{ij}\}$$

Given the angular information between vector pair (v_i, w_j) with $\langle v_i, w_j \rangle = k_{ij}$, we can conclude that:

$$E[\langle x, y \rangle] = \mu_X^T \mu_Y = \sum_{i,j}^{k,k} \lambda_i \tilde{\lambda}_j k_{ij}$$

We assume no concrete prior information on Σ_X, Σ_Y, treating them as a random orthonormal rotation matrix for now, and also, A and B are unrelated, meaning $\Sigma_{XY} = 0$.

To simplify calculations, we want to point back to the bilinear pairing in one attention row, as described in 6.1, ignoring the normalization factor $\frac{1}{d}$, as we restrict our embedding vectors onto spheres:

$$A_i = \begin{bmatrix} \langle W_q x_i, W_k x_1 \rangle & \langle W_q x_i, W_k x_2 \rangle & \dots & \langle W_q x_i, W_k x_n \rangle \end{bmatrix}$$

We observe that the same query-transformed vector $W_q x_i$ is paired with all key-transformed vectors $W_k x_1, \dots, W_k x_n$. By abstracting the query weight matrix W_q into the vectors, we can effectively consider the form $x_i^T C x_j$ with $C = W_k$ instead of $W_q^T W_k$. That would correspond to analyzing $\langle x, W_k y \rangle$, which is easier than $\langle W_q x, W_k y \rangle$. That brings the advantage that each element of C_{ij} is normally distributed rather than chi-squared. For the remainder of this subsection, we set $C_{ij} \sim \mathcal{N}\left(0, \frac{1}{n}\right)$ as a Xavier-initialized random matrix.

Now begins the confusing part. We have two sources of randomization: the probability distribution (hypersphere model) and the randomized weight matrix C. Thus, we need to determine the expected value of an expected value, with the outer sampling over the random matrix C and the inner over the data probability distribution.

$$\mathbb{E}[\mathbb{E}[x^T C y]] = \mathbb{E}[\mu_X^T C \mu_Y] = \mathbb{E}[\left(\sum_{i=1}^{k} \lambda_i v_i\right)^T C \left(\sum_{j=1}^{k} \tilde{\lambda}_j w_j\right)] = \sum_{i,j=1}^{k,k} \lambda_i \tilde{\lambda}_j \mathbb{E}[v_i^T C w_j] \tag{6.39}$$

The individual terms $v_i^T C w_i$ are decisive because they determine which angular information $\langle v_i, w_j \rangle$ is transmitted.

Proceeding further, with v_{il} denoting the l'th element of v_i, and considering $\mathbb{E}[C_{ij}] = 0$:

$$\mathbb{E}[v_i^T C w_j] = \sum_{l,m=1,1}^{n,n} \mathbb{E}[C_{lm} v_{il} w_{jm}] = \sum_{l,m=1,1}^{n,n} \mathbb{E}[C_{lm}] \mathbb{E}[v_{il}] \mathbb{E}[w_{jm}] = 0 \tag{6.40}$$

That implies:

$$\longrightarrow \mathbb{E}[\mathbb{E}[x^T C y]] = \sum_{i,j=1,1}^{k,k} \lambda_i \tilde{\lambda}_i \mathbb{E}[v_i^T C w_j] = 0 \tag{6.41}$$

Since $\mathbb{E}[x^T C y] = 0$, and given that C is a random matrix, that does not provide direct insights. It is not surprising that multiplying structural content with zero-centered noise results in an expected value of zero. However, does the variance retain that prior information instead?

What about $\text{Var}\left(\mathbb{E}[x^T C y]\right)$ (variance relates to the random matrix; the expected value to the Gaussians)?

$$\text{Var}\left(\mathbb{E}[x^T C y]\right) = \text{Var}\left(\mu_X^T C \mu_Y\right) \approx \sum_{i,j=1,1}^{k,k} \text{Var}\left(\lambda_i \tilde{\lambda_j} v_i^T C w_j\right) = \sum_{i,j=1,1}^{k,k} \lambda_i^2 \tilde{\lambda_j^2} \text{Var}\left(v_i^T C w_j\right) \tag{6.42}$$

With $\text{Var}\left(C_{ij}\right) = \frac{1}{n}$ and the matrix being independent of the vectors:

$$\text{Var}\left(v_i^T C w_j\right) = \sum_{l,m=1,1}^{n,n} \text{Var}\left(C_{lm} v_{il} w_{jm}\right) = \sum_{l,m=1,1}^{n,n} \frac{1}{n} \text{Var}\left(v_{il} w_{jm}\right) = \frac{1}{n} \sum_{l,m=1,1}^{n,n} \text{Var}\left(v_{il} w_{jm}\right) \tag{6.43}$$

Splitting into terms with equal and different embedding vector feature dimensions l and m (v_{il} denoted the l'th element of v_i):

$$\frac{1}{n} \sum_{l,m=1,1}^{n,n} \text{Var}\left(v_{il} w_{jm}\right) = \frac{1}{n} \sum_{l=1}^{n} \text{Var}\left(v_{il} w_{jl}\right) + \frac{1}{n} \sum_{l \neq m}^{n,n} \text{Var}\left(v_{il} w_{jm}\right) \tag{6.44}$$

Given $\langle v_i w_j \rangle = k_{ij}$ and referring to variance statistics 6.35 for hyperspheres gives us:

$$\forall l = m : \sum_{l=1}^{n} \text{Var}\left(v_{il} w_{jl}\right) = \mathbb{E}[v_i w_j]^2 = k_{ij}^2 \quad \forall l \neq m : \text{Var}\left(v_{il} w_{jm}\right) = \frac{1}{n} \frac{1}{n} = \frac{1}{n^2} \tag{6.45}$$

Inserting these calculations:

$$\frac{1}{n} \sum_{l=1}^{n} \text{Var}\left(v_{il} w_{jl}\right) + \frac{1}{n} \sum_{l \neq m}^{n,n} \text{Var}\left(v_{il} w_{jm}\right) = \frac{k_{ij}^2}{n} + \frac{1}{n} \sum_{l \neq m}^{n,n} \frac{1}{n^2} = \frac{k_{ij}^2}{n} + \frac{1}{n} \frac{n\left(n-1\right)}{n^2} \tag{6.46}$$

$$\frac{k_{ij}^2}{n} + \frac{1}{n} \frac{n\left(n-1\right)}{n^2} = \frac{k_{ij}^2}{n} + \frac{n-1}{n^2} \tag{6.47}$$

Combining each term together:

$$\text{Var}\left(\mathbb{E}[x^T C y]\right) = \sum_{i,j=1}^{k,k} \lambda_i^2 \tilde{\lambda_j^2} \text{Var}\left(v_i^T C w_j\right) \approx \sum_{i,j=1}^{k,k} \lambda_i^2 \tilde{\lambda_j^2} \left(\frac{k_{ij}^2}{n} + \frac{n-1}{n^2}\right) \tag{6.48}$$

That demonstrates that rotational or angular information is abstracted in the variance and preserved.

Combining this discovery with the Softmax formula from 5.20:

$$\mathbb{E}[\frac{p_i}{p_j}] = \exp\left(\frac{\mu_i - \mu_j + \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}}{2}\right)$$

We can see that $\sigma_i^2 + \sigma_j^2$ in the nominator of the exponential term raises when $\mathrm{Var}\left(\mathbb{E}[x^T C y]\right)$ increases with $\langle v_i, w_j \rangle = k_{ij}$, although this effect is rather subtle. The angle k_{ij} gets weighted by the linear combination factors $\lambda_i \lambda_j$ and contributes quadratically, starting to dominate $\frac{n-1}{n^2} \approx \frac{1}{n}$ for $k_{ij} > \frac{1}{n}$, which matches the variance magnitude frequently encountered throughout our analysis (Xavier initialization).

This long chain of reasoning can be bluntly summarized as analogous to multiplying noise with zero mean over images and exponentiating each pixel, which can reveal original patterns in the right value domain; the variance stores and memorizes uncertainty, which log-normally distributed variables also exhibit in their mean. Hence, it is interesting to conceptualize that the quadratic contribution is mathematically conceivable in bilinear forms for Transformer attention modules. In retrospect, utilizing Dirichlet distributions would have been more elegant in abstracting the hypersphere models, which marks an interesting path to pursue.

6.3.3 Upper Scaling Limit for Embedding Matching

The Cauchy-Schwartz inequality provides us the formula, using $||\cdot||_\infty$ to denote the matrix operator norm (equivalent to the largest singular value) of a matrix, thus bounding the bilinear from above:

Statement: For $x, y \in \mathbb{R}^n$ and $W_q, W_k \in \mathbb{R}^{n \times n}$:

$$|\langle W_q x, W_k y \rangle| \quad \leq \quad ||W_q||_\infty \cdot ||W_k||_\infty \cdot ||x||_2 \cdot ||y||_2 \cdot |\cos\left(\angle\left(x, y\right)\right)|$$

Proof: The matrix operator norm $||\cdot||_\infty$ is sub-multiplicative:

$$||W_q^T W_k||_\infty \leq ||W_q^T||_\infty \cdot ||W_k||_\infty = ||W_q||_\infty \cdot ||W_k||_\infty \tag{6.49}$$

And using $|\langle x, y \rangle| = ||x||_2 \cdot ||y||_2 \cdot \|cos(\angle\left(x, y\right))|$ concludes the proof.

In terms of scaling behavior, this means that the length of a scaled vector is unlikely to exceed a factor of $2 \cdot 2 = 4$ for two random matrices W_q and W_k with Xavier initialization $\mathcal{N}\left(0, \frac{1}{n}\right)$. This is because we have already shown that the squared largest singular value is close to $4n\sigma^2 = 4n\frac{1}{n} = 4$ in that case, making 2 the approximated largest singular value, and that is only for perfect alignment.

That gives the model opportunities to allow important vector pairing to set themselves apart. Random vectors are expected to be dissimilar to most learned embedding vectors due to property 5.3. And token embeddings x and y, which are nearly orthogonal on the sphere $\cos\left(\angle\left(x, y\right)\right) \approx 0$, have a lower chance of attending to themselves since the largest singular value of both A and B is limited to 2 in random matrices (the singular values are always samples from the same Wigner spectrum as illustrated in the function 2.4). Therefore, the expression $\langle Ax, By\rangle$ remains small. On the contrary, linguistically related embeddings with $\cos\left(\angle\left(x, y\right)\right) \approx 1$ must be actively suppressed by being aligned to small singular values and making them more likely to attend to themselves.

That also provides insight into how models perceive linear combinations or how they can distinguish terms more easily. In Transformers, positional encodings $\hat{x}_i$ are often added to the embedding vectors to identify their position: $\langle W_q\left(x + \hat{x}\right), W_k\left(y + \hat{y}\right)\rangle = \langle W_q x, W_k y\rangle + \langle W_q \hat{x}, W_k y\rangle + \langle W_q x, W_k \hat{y}\rangle + \langle W_q \hat{x}, W_k \hat{y}\rangle$. Positional encodings are often dissimilar to most embedding vectors due to their sinusoidal nature, so $\langle x, \hat{y}\rangle$ and $\langle \hat{x}, y\rangle$ become small. Consequently, the mixed terms $\langle W_q \hat{x}, W_k y\rangle$ and $\langle W_q x, W_k \hat{y}\rangle$ get suppressed due to the Cauchy-Schwartz inequality and $\langle \hat{x}, \hat{y}\rangle$ reaches higher values for neighboring tokens.

6.4 Is the result of the Bilinear Form normally distributed?

As a short explanation of whether log-normal distributions in the SoftMax chapter are justified, the distribution is not normal due to the following reasoning:

$$\langle Ax, By\rangle = x^T A^T B x = x^T C y = \sum_{i,j=1}^{n,n} C_{ij} x_i y_i$$

The product $x_i y_i$ is chi-squared distributed, alike $C_{ij} = \sum_{k=1}^{n} A_{ik} B_{ki}$. The sum of chi-squared distributions is again chi-squared distributions with the aggregated sum of the degree of freedoms of its components. However, the product of two chi-squared variables is only chi-squared when independent, which is certainly not the case for coupled learned weight matrices but is still applicable for a theoretical investigation. Finally, utilizing the Central Limit Theorem (CLT) is recommended, as chi-squared distribution approaches a normal distribution for $n \to \infty$ without additional normalization steps. That still justifies, to some extent, the log-normal distribution after the exponentiation in the Softmax operator, motivated by the vast amount of chi-squared distributed variables entering altogether.

6.5 Summary Attention as Bilinear Form with Randomized and Learned Weight Matrices

This section provides insights into how the attention weight matrices $A = W_k$ and $B = W_q$ relate to each other in the bilinear form $\langle Ax, By \rangle$ and how they interact with the probability distributions p_A and p_B. The first two statistical moments of the bilinear form $\langle Ax, By \rangle$, are particularly important, although they are often challenging to determine due to the complexity of the chi-squared distributions. In the case of p_A and p_B being Gaussians, we discovered that the base alignment matrix $C = A^T B$ and its trace plays a key role in how attention between the same embedding vectors $\langle Ax, Bx \rangle$ and different dependent and independent embedding vectors $\langle Ax, By \rangle$ is upscaled. We focused on randomized matrices and their respective singular value spectrum to determine how expected value and variance can be determined. We also examined the utility and limitations of singular value distributions for attention weight matrices. Furthermore, we explored how pre-trained angular embedding information $\langle x, y \rangle$ of linear combinations is preserved. By modeling hyperspheres, we found that even for random matrices, this information is not lost due to the higher variance of the associated terms, which is essential for the attention scores in Softmax, emphasizing the significance of the statistical moments of the bilinear form for the subsequent processing in the attention mechanism. We also estimated an upper bound on embedding pairings, indicating that random embedding vectors are less likely to disturb attention patterns.

7 Gaussians Mixture Models and the "Uniform Embedding Mixture Problem"

7.1 Linear Combinations of Gaussians in Attention

After examining the query and key weight matrix W_q and W_k, we now get to the value weight matrix W_v and the linear combinations of embeddings obtained after calculating and multiplying with the attention matrix $A \in \mathbb{R}^{n \times n}$.

7.1.1 Gaussian Mixture Models in Attention

The original data distribution matrix has the following format:

$$\mu_X = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_n \end{bmatrix} \quad \Sigma_X = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{n1} & \dots & \dots & \Sigma_{nn} \end{bmatrix}$$

The value weight matrix W_v does not alter linear combinations $\sum_{i=1}^{n} \lambda_i \cdot v_i$, like all other linear layer weight matrices. Usually, bias terms are not used in attention-weight matrices but in the subsequent feedforward layers.

As we now operate in the value embedding domain after the applied Softmax function, we apply the change of base through W_v as illustrated in 4.2, so that VX becomes our new Joint Gaussian.

After applying row-wise Softmax, we get a linear combination $\begin{bmatrix} \lambda_1^{(j)} & \lambda_2^{(j)} & \dots & \lambda_n^{(j)} \end{bmatrix}^T$ for each row j in the normalized attention matrix A, with $0 \leq \lambda_i^{(j)} \leq 1, \sum_{i=1}^{n} \lambda_i^{(j)} = 1$. Remembering the formula $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, we can rewrite the statistics.

The resulting embedding at position j becomes a linear combination of all previous value embeddings scaled by the respective attention score $\lambda_i^{(j)} = A_{ij}$ with $\hat{X}_j = \sum_{i=1}^{n} \lambda_i^{(j)} X_i$, resulting in numerous probability mixture models, specifically, Gaussian mixture models. The new joint Gaussian $\mathcal{N}(\mu_{\hat{X}}, \Sigma_{\hat{X}})$ is specified by:

$$\mu_{\hat{X}} = \begin{bmatrix} \sum_{i=1}^{n} \lambda_i^{(1)} \mu_i \\ \sum_{i=1}^{n} \lambda_i^{(2)} \mu_i \\ \dots \\ \sum_{i=1}^{n} \lambda_i^{(n)} \mu_i \end{bmatrix}$$

And by using (recalling $\text{Var}\left(\hat{X}_k\right) = \text{Cov}[\hat{X}_k \hat{X}_k]$)

$$\text{Cov}[\hat{X}_k \hat{X}_l] = \text{Cov}[\sum_{i=1}^{n} \lambda_i^{(k)} X_i, \sum_{j=1}^{n} \lambda_j^{(l)} X_j] = \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_i^{(l)} \text{Cov}[X_i, X_j] = \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_i^{(l)} \Sigma_{ij}$$

, we obtain the new joint covariance matrix of $\hat{X}$:

$$\Sigma_{\hat{X}} = \begin{bmatrix} \sum_{i,j=1}^{n,n} \lambda_i^{(1)} \lambda_j^{(1)} \Sigma_{ij} & \sum_{i,j=1}^{n,n} \lambda_i^{(1)} \lambda_j^{(2)} \Sigma_{ij} & \dots & \sum_{i,v=1}^{n,n} \lambda_i^{(1)} \lambda_j^{(n)} \Sigma_{ij} \\ \sum_{i,j=1}^{n,n} \lambda_i^{(2)} \lambda_j^{(1)} \Sigma_{ij} & \sum_{i,j=1}^{n,n} \lambda_i^{(2)} \lambda_j^{(2)} \Sigma_{ij} & \dots & \sum_{i,v=1}^{n,n} \lambda_i^{(2)} \lambda_j^{(n)} \Sigma_{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i,j=1}^{n,n} \lambda_i^{(n)} \lambda_j^{(1)} \Sigma_{ij} & \dots & \dots & \sum_{i,v=1}^{n,n} \lambda_i^{(n)} \lambda_j^{(n)} \Sigma_{ij} \end{bmatrix}$$

The new cross-covariance matrices are a weighted sum of all previous cross-covariance matrices, scaled by the corresponding contribution values $\lambda_i^{(k)}$ and $\lambda_j^{(l)}$ from the rows k and l. Theoretically, these new cross-covariance terms are additionally influenced by the interdependence of the covariance matrices, which is determined by the fourth moment of the Gaussian-approximated data probability distribution, which we ignore for obvious reasons.

7.1.2 Information Theory on Gaussian Mixture Models

The new Entropy $\text{H}\left(\hat{X}_k\right)$ and Mutual Information values $\text{I}\left(\hat{X}_k; \hat{X}_l\right)$ become:

$$\text{H}\left(\hat{X}_k\right) = \frac{1}{2} \log\left((2\pi e)^d \left| \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_j^{(k)} \Sigma_{ij} \right|\right)$$

$$\text{I}\left(\hat{X}_k; \hat{X}_l\right) = \frac{1}{2} \log \frac{\left| \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_j^{(k)} \Sigma_{ij} \right| \left| \sum_{i,j=1}^{n,n} \lambda_i^{(l)} \lambda_j^{(l)} \Sigma_{ij} \right|}{|\Sigma_{\hat{X}_{kl}}|}, \text{with}$$

$$\Sigma_{\hat{X}_{kl}} = \begin{bmatrix} \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_j^{(k)} \Sigma_{ij} & \sum_{i,j=1}^{n,n} \lambda_i^{(k)} \lambda_j^{(l)} \Sigma_{ij} \\ \sum_{i,j=1}^{n,n} \lambda_i^{(l)} \lambda_j^{(k)} \Sigma_{ij} & \sum_{i,j=1}^{n,n} \lambda_i^{(l)} \lambda_j^{(l)} \Sigma_{ij} \end{bmatrix}$$

Breaking down becomes obviously considerably more complicated, as the determinant is not linear. But we can see that the Mutual Information $\mathrm{I}\left(\hat{X}_k; \hat{X}_l\right)$ is not preserved since the model shifted information amongst the marginal embedding distributions.

7.1.3 Gaussian Mixture Models Processing in Transformers

These linear combined Gaussian Mixture Models now get into small feed-forward neural networks in the following order, such as illustrated in the plot 2.2:

$$y = \text{LayerNorm}\left(\text{RELU}\left(\text{FFN}_{\text{Up}}\left(\text{Attention}\left(Q, K, W, x\right) + \text{SKIP}\left(x\right)\right)\right)\right)$$

$$z = \text{LayerNorm}\left(\text{RELU}\left(\text{FFN}_{\text{Down}}\left(y\right)\right)\right)$$

In general, FFN_{Up} increases the embedding dimensionality significantly, while FFN_{Down} scales it down the original dimension. We already mentioned that linear maps do not alter the probability distribution map of the joint Gaussian $\mathcal{N}\left(\mu_{\hat{X}}, \Sigma_{\hat{X}}\right)$ in terms of the linear contributions factors $\lambda_i^{(l)}$, so only RELU, LayerNorm, and SKIP are left to impact them.

It is possible to propagate the statistical moments through RELU [9], but that is complicated, and I simply proceed without this, but the linear combinations get partly broken.

Regardless, the LayerNorm is interesting, as it normalizes each embedding component $\hat{X}$ to have zero mean and a variance of 1 in each dimension. That could be used for the hypersphere radius l in the SoftMax function as $\langle Ax, By \rangle \sim ||x||_2, ||y||_2$ and so does this affect the distribution of $\langle W_q x, W_k y \rangle$. In case the matrices trace product behaves like for randomized matrices with $\text{Tr}\left(W_q^T W_k\right) \approx \mathcal{N}\left(0, 1\right)$, it is to assume that the distribution of $\langle W_q x, W_k y \rangle$ shares commonalities with that of the Xavier initialization $\mathcal{N}\left(0, l \cdot \frac{1}{n}\right)$ with hypersphere radius $l = 1$ due to equation 6.2.

Concerning the SKIP connection, if the matrix singular spectrum of the value weight matrix W_v is known, it is possible to integrate the previous Gaussian mixture model into the current Gaussian Mixture Model, which gets aligned

due to the Layer Normalization by a linear combination of the current Gaussian Mixture Model and that of the previous layer.

7.2 Flat Mixture Distribution Problem: How can Transformers distinguish embeddings?

7.2.1 Attention is Low-Rank

In their seminal work, Wang et al. [46] proved an important theorem in the "Linformer" paper, establishing that attention is a low-rank operator. This theorem, translated into our notation, is as follows:

Theorem: For any embedding sequence $X \in \mathbb{R}^{n \times d}$ and matrices $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$, and for any column vector $w \in \mathbb{R}^n$ of matrix $W_v X$, there exists a low-rank matrix $\tilde{P} \in \mathbb{R}^{n \times n}$, such that $\Pr\left(||\tilde{P}w^T - Pw^T|| < \epsilon||Pw^T||\right) > 1 - o(1) \approx 1 - 2e^{-\left(\epsilon^2+\epsilon^3\right)\frac{\log(n)}{4}}$ and $\operatorname{rank}\left(\tilde{P}\right) \in \Theta(\log(n))$, with $P = \text{Softmax}\left(\frac{(W_q X)^T (W_k X)}{\sqrt{d_k}}\right)$.

The theorem concludes that the rank of $P = \text{Softmax}\left(\frac{(W_q X)^T (W_k X)}{\sqrt{d_k}}\right) \in \mathbb{R}^{n \times n}$ is approximately $\Theta(\log(n))$, indicating that the remaining singular values are close to zero. Although the detailed reasoning is beyond the scope of this work, intuitively, this stems from the observation that the exponentiated Gram Matrix $= (W_q X)^T (W_k X) = Q^T K$ is highly compressible in high-dimensional space due to the non-linear asymmetry introduced by exponentiation concerning the Euclidean distance in the original space.

This sparsity is consistent with the low rank of the attention weight matrices W_q and W_k. Despite their low dimensionality, translating this into a low-rank functional form is complex. Specifically, although each attention operation Softmax $\left(\frac{(W_q X)^T (W_k X)}{\sqrt{d_k}}\right)$ can be realized with a low rank, achieving a tight upper bound for the rank of the entire operator is challenging. The "Linformer" paper and the work by Qin et al. [32] address the discrepancy. In the theory section 4.3, we utilized prior embedding information to reduce the rank of the Gaussian covariance matrices. Similarly, this approach can be applied to the learned weight matrices, effectively managing the second random component so that we can effectively contrast two low-rank entities.

7.2.2 Sparse Singular Value Spectrum Problem

At this point, I must briefly refer to my experimental section. My experiments (visualized in the Appendix section A.5) demonstrated that the singular value spectrum (with Xavier initialization for non-quadratic matrices) is pressed down, making the matrix more sparse by contrasting the initial state in Figures A.19 with the learned state A.20 for attention weight matrices and A.21 normal feedforward layers. The base alignment $W_q^T W_k$ in diagram A.22 behaves like the product of random matrices with $\mathbb{N}^{\langle}(0, \sigma^2\rangle)$ (section 6.2.3) but is relatively unstable in this regard.

Let $\sigma'_1, \sigma'_2, \ldots, \sigma'_n$ be the descendingly ordered singular values of the learned weight matrix $\hat{A}$. I observed the following for W_q and W_k

$$\mathbb{E}[\sum_{i=1}^{n} \hat{\sigma_i}^2] \approx \frac{n}{20} - \frac{n}{10} < n = \mathbb{E}[\sum_{i=1}^{n} \sigma_i^2] \quad \mathbb{E}[max\,(\hat{\sigma_i})] \approx 3 - 4 > 2 = \mathbb{E}[\max(\sigma_i)]$$

And for W_v:

$$\mathbb{E}[\sum_{i=1}^{n} \hat{\sigma_i}^2] \approx \frac{9n}{10} - \frac{12n}{10} \approx n = \mathbb{E}[\sum_{i=1}^{n} \sigma_i^2] \quad \mathbb{E}[max\,(\hat{\sigma_i})] \approx 2 - 3 > 2 = \mathbb{E}[\max(\sigma_i)]$$

Therefore, a random vector is scaled down to about a tenth of its original length on average. I observed that for the largest $\approx 10\%$ singular values: $\sigma'_i > \sigma_i$. However, a vector, primarily orthogonal to the largest principal components, is substantially upscaled compared to a random-orientated counterpart. Thus, the learned principal components of V_Q and V_K are of importance, with the possibility to maximize $\langle Qx, Ky\rangle$ for the corresponding alignment of orthogonal subspaces from V_Q and V_K. It is important to note that U_Q and U_K can reverse that effect, but the asymmetry in the singular value spectrum breaks many structurally preserving properties of random matrices. Consequently, the scaling should now emphasize dominant orthogonal linear subspaces.

However, the base alignment $W_q^T W_k$ in diagram A.22 behaves like the product of random matrices with $\mathcal{N}\left(0, \sigma^2\right)$ (section 6.2.3) for most layers l, with $\mathrm{Tr}\left(W_q^{(l)} W_k^{(l)}\right) \sim \mathcal{N}\left(0, 1\right)$ instead of $\mathrm{Tr}\left(W_q^{(l)} W_k^{(l)}\right) \approx n$ for aligned matrices. This implies that the query and key bases $Q = \begin{bmatrix} | & | & & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \\ | & | & & | \end{bmatrix}^T$ and $K =$

$\begin{bmatrix} | & | & & | \\ \mathbf{k}_1 & \mathbf{k}_2 & \cdots & \mathbf{k}_n \\ | & | & & | \end{bmatrix}$ are aligned like uncorrelated random matrices. That implies that the scaling of the embeddings is, on average, operated in different singular vector bases, and prior angular information is barely preserved. That does not imply that there is no preservation, as contradicted by the John Lindenstrauß Lemma and the sparse operator functional. Still, it makes the angular processing behavior subtle and difficult to interpret.

7.2.3 The Uniform Embedding Mixture Problem

We can now formulate a problem inherent in Transformer models, which I want to label a "Uniform Embedding Mixture Problem." Can the model understand the linear combinations of the Gaussian Mixture model, and how can it still discern token embeddings after multiple Softmax passes?

During training, the singular value spectra of query and key embeddings become sparse. Combined with the apparently randomized alignment of the bases in W_q and W_k, it causes the bilinear form $\langle W_q x, W_k y \rangle$ to press the vectors onto a hypersphere with a smaller radius $l < 1$. As a result, the probability distributions after Softmax become quite flat, with the relational factor of log-normally distributed variables $\mathbb{E}[\frac{\exp(\langle W_q x_i, W_k x_j \rangle)}{\exp(\langle W_q x_i, W_k x_l \rangle)}]$ close to 1. That implies that the linear combination of value embeddings $V_1, \ldots, V_n$ have almost equal weighting $\lambda_1 \ldots, \lambda_n \approx \frac{1}{n}$, and their Softmax-transformed counterpart $\text{Softmax}\left(\begin{bmatrix} \lambda_1 & \ldots & \lambda_n \end{bmatrix}^T\right)$ approaches maximum entropy $\log(n)$ The distortion effects by the cross-covariance matrices Σ_{ij} between the X_i are thus equally weighted. In the subsequent layer, it remains questionable how the model can dissect the Mixture Embedding into its components, as the flatness effects on linear combinations are perpetuated and exacerbated throughout the network, as all these patterns repeat themselves.

The following statement formulates the "Uniform Embedding Mixture Problem" inspired by the "token similarity problem"[47] under Gaussian approximation of the input data, disregarding Skip connections and non-linear RELU transformations.

Hypothesis:

Let $X_1, \ldots, X_n$ denote a series of embedding initially distributed by a joint Gaussian such as described in equation 4.1. Let $l \in \mathbb{N}$ denote a layer l with intermediate results $X_1^{(l)}, \ldots, X_n^{(l)} = \text{Self-Attention}\left(W_q^{(l)}, W_k^{(l)}, W_v^{(l)}, \left(X_1^{(l-1)}, \ldots, X_n^{(l-1)}\right)\right)$, where $W_q^{(l)}, W_k^{(l)}, W_v^{(l)}$ are weight matrices with respective operator norm $||W_q^{(l)}||_\infty \leq q_{max} \leq 2, ||W_k^{(l)}||_\infty \leq k_{max} \leq 2, ||W_v^{(l)}||_\infty \leq v_{max} \leq 2$. Let $p_i^{(l)}$ denote the probability distribution of $X_i^{(l)}$. Then, for $l \to \infty$, there exists a probability distribution p such that $\forall i \in 1, \ldots, n : p_i^{(l)} \to p$

This statement appears true for the special case of random matrices, as the linear combination factors λ_i in the Gaussian mixture components have, as stated in equations 5.1, 5.2 an almost equal contribution factor. Applying the formula for the ratio of log-normal variables 5.20, assuming that all probability distributions merge, seems reasonable. The challenge involves calculating all the statistics of the bilinear form, such as stated in 6.2, and progressing further. An interesting question is how the spectral properties q_{max}, k_{max}, and particularly v_{max} influence the statement and how it relates to learned matrices of $\text{rank}\,(W_{q,k,v}) \in \Theta\,(\log\,(n))$.

A more advanced form of the statement would include the skip connections $X_1^{(l)}, \ldots, X_n^{(l)} = \beta \cdot \text{Self-Attention}\left(W_q^{(l)}, W_k^{(l)}, W_v^{(l)}, \left(X_1^{(l-1)}, \ldots, X_n^{(l-1)}\right)\right) + (1-\beta) \cdot X_1^{(l)}, \ldots, X_n^{(l)}$ to investigate whether all individual distributions still converge to the same one ($\beta \in [0, 1]$ dependent on singular value spectrum of W_v to simulate Layer Normalization). This approach would elucidate the dynamics of Transformer models, but it tremendously complicates the calculations.

7.3 Summary Gaussians Mixture Models and the "Uniform Embedding Mixture Problem"

Assuming that all data is jointly Gaussian-distributed, each embedding, after one attention operation, becomes a linear combination of embeddings or, more generally, a Mixture of Gaussian distributions. This poses questions about how models can handle these linear combinations and how they can dissect them. The "Uniform Embedding Mixture Problem" hypothesis addresses the issue that log-normally distributed variables result in relatively flat probability distribution mixtures with equal shares. This hypothesis is crucial for understanding advanced Transformer architectures' inference dynamics and modeling capacities.

Examining these effects, especially concerning the singular value spectra of randomized and learned weight matrices, is challenging. This requires evaluating the joint Gaussian distribution, Softmax geometry, and bilinear form statistics as a whole, both for randomized and learned matrices. This involves extensive calculations, including determining which parameters to fix to make further progress. Additionally, incorporating non-linear ReLU and crucial Skip connections into this theoretical framework is essential.

It is also interesting to think about Anthropic's discussions on polysemanticity [13] concept, [12], which relate to the problem of how intermediate activation patterns can exhibit a superposition of multiple feature representations. The insightful assumption is that a Transformer model operates on a higher-dimensional basis, which is then compressed and can be regained via Dictionary Learning from an Autoencoder model. The information density observed during the inspection of the bilinear form makes further interpretations challenging and conclusions difficult to draw, but the Mixture Model conceptualization advocates the superposition hypothesis [12].

One potential approach to addressing the "Uniform Embedding Mixture Problem" is through the "residual stream" model [13], which proposes communication layers across various layers. This model suggests that information is shared beyond the processing of both non-linear ReLU and attention modules. This leads to the conclusion that mixture models should always include a component pointing to themselves, which could be crucial for proving or refuting the "Uniform Embedding Mixture Problem" hypothesis.

8 Experiments: Attention in Few-Shot Prompting

8.1 Model, Dataset, and Task Objective

Following the extensive theoretical investigation, this experimental section tests some of the theoretical findings. Originally, it primarily generated theoretical ideas based on practical findings, in contrast to the usual order of empirical Machine Learning research. The main objective of this study is to advance the theory of Transformer models under Gaussian approximation through logical reasoning and proofs rather than extensive practical evaluation, which would be the next step in expanding this research.

Model Specifics. The experimental setup centers around 2023 released Transformer model "Mistral-7B-Instruct-v0.2" [18]. This model was chosen for its recent state-of-the-art performance, comparable to LLaMa 2 [41], ensuring that the findings are sufficiently robust while managing storage requirements. The Mistral-7B model employs Sliding Window Attention, altering the attention functionality by attending to only the previous m tokens. Here, all tokens in the input sequence attend to all previous tokens, but only the last m tokens are effectively considered by the model during training. Additionally, Group Query Attention (GQA) (see Figure 8.1) aggregates multiple queries with one key to enhance performance [1].

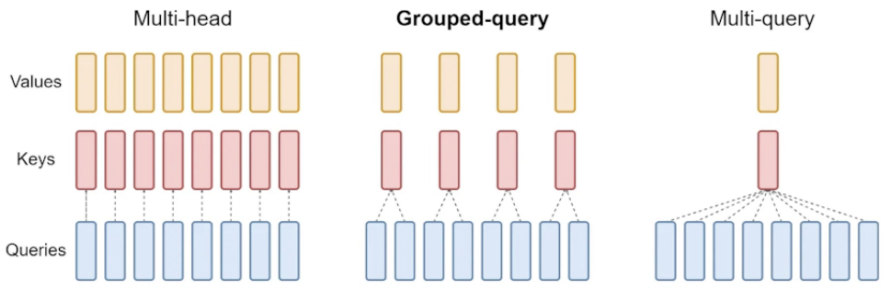


Fig. 8.1.: Group Query Attention

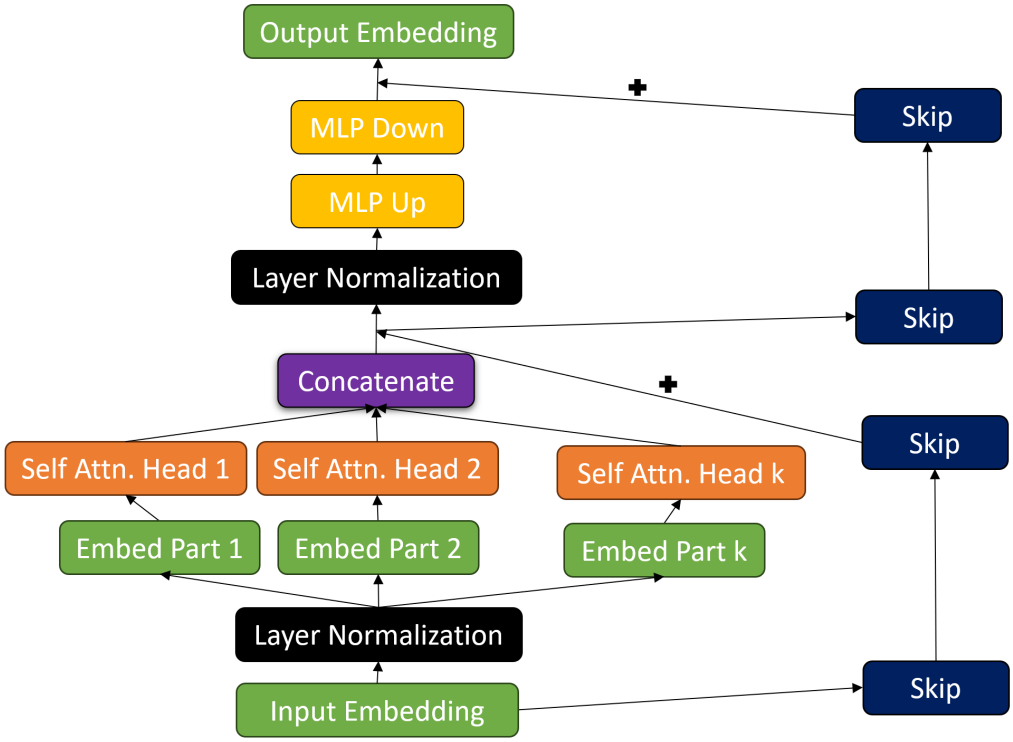


Fig. 8.2.: Mistral-7B Transformer Block Schema

The Mistral-7B model, like many recent large language models, follows a decoder-only approach, which means the attention mechanism functions slightly differently from the original "Attention is All You Need" model [43], as the encoding is already realized in the pre-trained embedding. Despite these changes, the architecture of a Transformer Bock is similar to LLaMa [40], as depicted in Figure 8.2. The Mistral-7B model comprises 32 transformer blocks with 32 attention heads each. The vocabulary size is 32000, and the embedding size is 4096 for each transformer's input and output layers.

An additional output matrix W_o, widely established in LLaMa models [43, 40], complicates calculations. The output matrix W_o works as described in the transformer background section 2.3. In my experiments, all attention maps from all individual attention heads are averaged, simulating a single-head attention setting. However, weighting by the W_o parametrization is certainly more precise for further experiments.

Technical Setup & Storage Problems. From a technical perspective, I used an AMD Ryzen 5900X CPU and an NVIDIA 4070 Ti Super GPU with 16 GB VRAM. Managing storage with a 16 GB VRAM graphics card proved challenging and significantly hindered research progress. According to the author's

huggingface homepage, a 16-float version requires nearly 14 GB of GPU storage when fully loaded, allowing for inference but leaving only about 2 GB VRAM during inference. This often caused Jupyter Notebook kernel crashes when accumulating internal results, necessitating frequent restarts and most results being collected over multiple attempts. Although it is possible to load the model in 4-bit/8-bit quantization, translating all attention weights into a compressed format that is not easily reversible or interpretable forced me to store the weights separately in 16-bit quantization for analysis. For inference, I used the load-in-8-bit option to allocate less VRAM storage. However, by auto-loading, I frequently encountered bugs related to missing metadata in tensors, which were difficult to fix manually.

Task Objective. After discussing the technical challenges, we now turn to the task and dataset applied in the experiments. Initially, I analyzed attention maps on smaller models like BERT-uncased [10] and T5 [33], which yielded decent results. However, I tested Few-Shot Prompting, an in-context learning technique that has gained popularity in NLP research. This technique involves providing the model with a few examples (shots) to help it understand the expected answer format in terms of length and response style to generate a new answer.

The chosen task originated from an idea developed during my two-week NLP lab course in April 2024. We implemented a Crossword Generator and Automatic Solver for a specific topic, where I was responsible for developing a crossword clue generator for a list of words related to a specified category. This was my first application of Mistral 7B, deploying few-shot prompting to generate intriguing clues for each word, where I implemented the entire model pipeline myself under the supervision of a tutor. During the NLP lab course, we used crossword clues from the New York Times as training data, but they were often hard to understand due to abstract wording, so the newly generated clues were barely solvable. Later, while working on my thesis, I found the attention values for this few-shot prompting scenario quite interesting and decided to expand this model to this new use case with different data.

Generation of Toy Data. Regarding the data, I took an unconventional approach and generated it myself. I found that LLM models were adept at generating clues for words at the appropriate difficulty level. I prompted GPT-4 from OpenAI to generate 5000 clues divided into ten predefined categories.

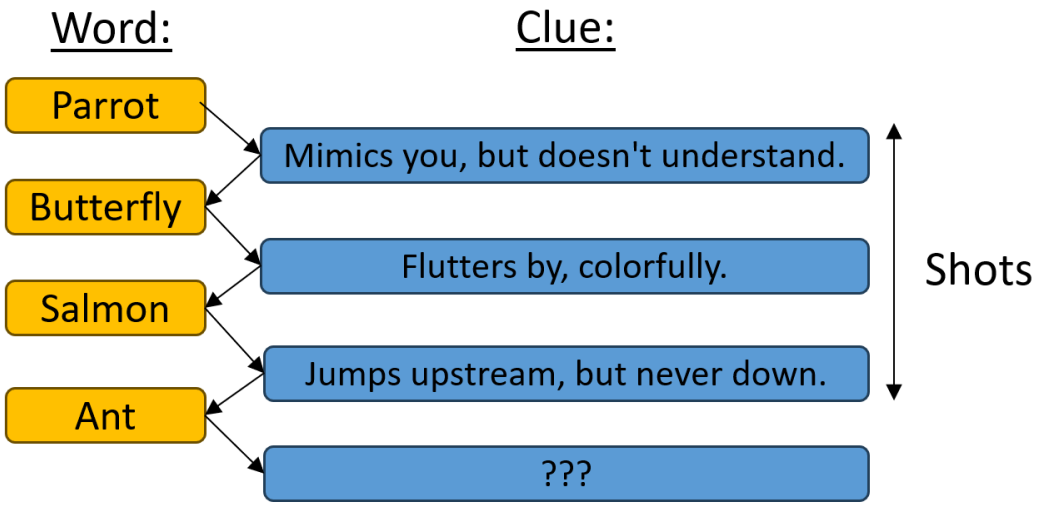


Fig. 8.3.: Few-Shot Prompting Clue Generation for $k = 3$ shots

The task itself is illustrated in Figure 8.3. As visualized, we pass a collection of k shots/examples with corresponding clues, and then the model generates a new clue in the style of the previous answers for the last word. The sentences are passed as shown in this example prompt:

"[INST] You are a crossword clue generator and have to generate a short clue for each input word from the category Animals*. I will give you a word, and you have to output a brief description of the word, which will be engaging and challenging. Just give one answer, marking the end with a #. [/INST] Ok [INST] Word: Parrot [/INST]Mimics you, but doesn't understand.# [INST] Word: Butterfly [/INST]Flutters by, colorfully.# [INST] Word: Salmon [/INST]Jumps upstream, but never down.# [INST] Word: Ant [/INST]"*

The prompt begins with a general context for the in-context learning framework, which is applied to all subsequent inquiries to shape the model's answers. The context ends with "Ok," followed by the randomized words and clues from the training dataset, separated by the token *"[INST]*. The model is tasked with generating a clue for the last word.

This concatenated sequence is tokenized and passed to the model, processed altogether, with the attention scores returned for each token pair in the entire prompt. The model has been pre-trained and is not fine-tuned, so we can consider Few-Shot Prompting as an inference on a trained model, where each few-shot operation is independent of the others, with the model not memorizing them.

Word	Generated Clue
Ant	Small, yet mighty workers in a complex society.
Bee	Honey-producing insect.
Fox	Cunning, red-hued forest dweller.
Raven	Large, black bird with a complex call.
Tarantula	Large, hairy, venomous arachnid.
Lemming	Small, rodent-like Arctic dweller.
Mink	Furred, semi-aquatic mammal.
Dog	Man's best friend.
Hare	Famous relative of the rabbit.
Hippopotamus	Large, semi-aquatic mammal with massive jaws.

Tab. 8.1.: Generated clues (3 shots) for target words of the category *Animals*

A list of example-generated answers for few-shot prompts for the category Animals can be seen for different numbers of shots in the Appendix A.1. Still, a small subset is depicted in Table 8.1. Interestingly, the quality and preciseness of the generated answer increase with a larger shot count k as the model actively learns the expected answer style and format.

8.2 Attention Maps Patterns

The attention maps are averaged over all 32 attention heads in a specific layer (transformer block). After applying the row-wise Softmax function, the results lie within the probability range $[0, 1]$. I chose to plot only the attention scores that address the words and clues; therefore, they do not sum to 1, as the tokens also attend to other tokens in the context and the separator token *"[INST]"*, which is not interesting for us. I fixed the random seed in my implementation to render the results reproducible.

Figure 8.4 illustrates an example without token aggregation, while Figure 8.5 shows the same example with subword token aggregation for chosen layers. Token aggregation involves summing all individual pair-wise attention scores between all combinations of subword tokens to maintain the total sum so that only the attention scores for entire words are visualized.

To explain the attention maps, the words on the left side attend to the words at the bottom, which are both the same sentence. Since Mistral7B only allows

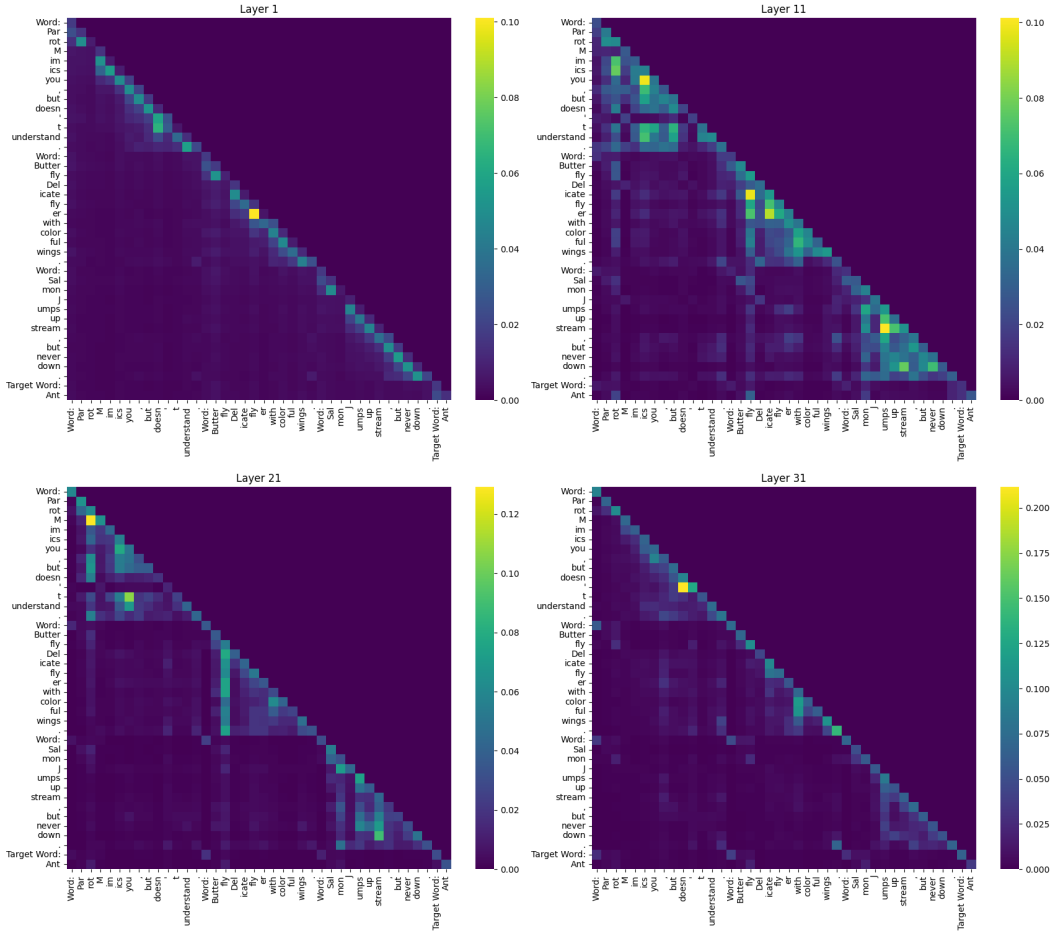


Fig. 8.4.: Attention Maps Token Level for Target Word *"Ant"*, with $(k = 3)$ shots," Generated Clue: *"Organized worker, colony builder."*

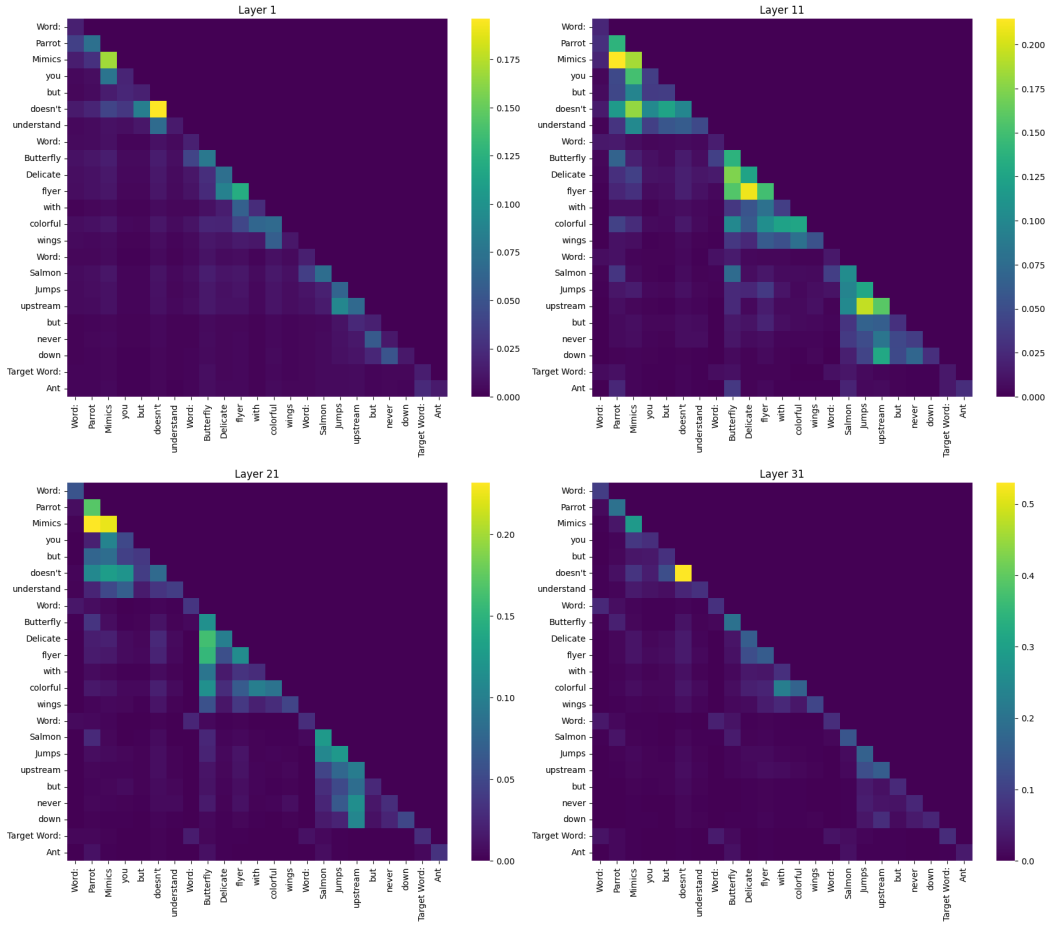


Fig. 8.5.: Attention Maps Word Level for Target Word *"Ant"*, with $(k = 3)$ shots," Generated Clue: *"Organized worker, colony builder."*

attention to previous tokens and itself, the matrix becomes lower-triangular; to the right and the bottom of each attention map are the depicted tokens of interest with their respective attention scores. The order on the left and bottom is the following:

$$\text{Word} \rightarrow \text{Example Clue} \rightarrow \text{Word} \rightarrow \text{Example Clue} \rightarrow \cdots \rightarrow \text{Target Word} \rightarrow$$

A complete pass A.2 showing all attention layers and additional illustrations for other words A.3 and a larger shot number k A.4 are all available in the Appendix.

Evolving Attention Patterns. The subdivided triangular structure in the lower-triangular attention maps from the averaged attention heads indicates that the model primarily attends within the context of a provided shot/example, separated by the *"[/INST]"* token. As expected, the attention map structure evolves across the attention layers (compare full pass A.2). Initially, the first attention maps are nearly diagonal-dominant, followed by increasing attention between terms at a fair distance, manifesting in the subdivided triangles. However, in the last layers, the attention patterns become sparser and resemble the diagonal-dominant matrices from the lower layers, aligning with findings from BERT interpretability research [8, 27, 20]. My presumption is that the model has already mixed the probability distribution into a desired "mixture state."

Linguistic Dependencies. It is noteworthy that certain linguistically related words attend to each other, such as *"mimics"* strongly attending to *"parrot"*. Many words in an example clue sequence attend to the given sample word of the shot. Generally, words do not attend across the boundaries of different shots, indicating that different training shots do not interfere with each other. This is remarkable because, although the entire sequence is passed to the model and marked by a separator token, the attention maps can perform some logical reasoning regarding language features. Despite LLaMa-related models being among the best, attention maps are still carefully considered, and the exact scoring often provides limited insight into the model's reasoning process. This may either require a larger streamlined model or mean that the attention mechanism cannot be concretely explained and that most linguistic learning occurs in other network parts, which is a valid concern.

Value Domain. Examining the value domain of the attention scores in the attention map on token level 8.4, we see they range from $[0, 0.1]$. This suggests

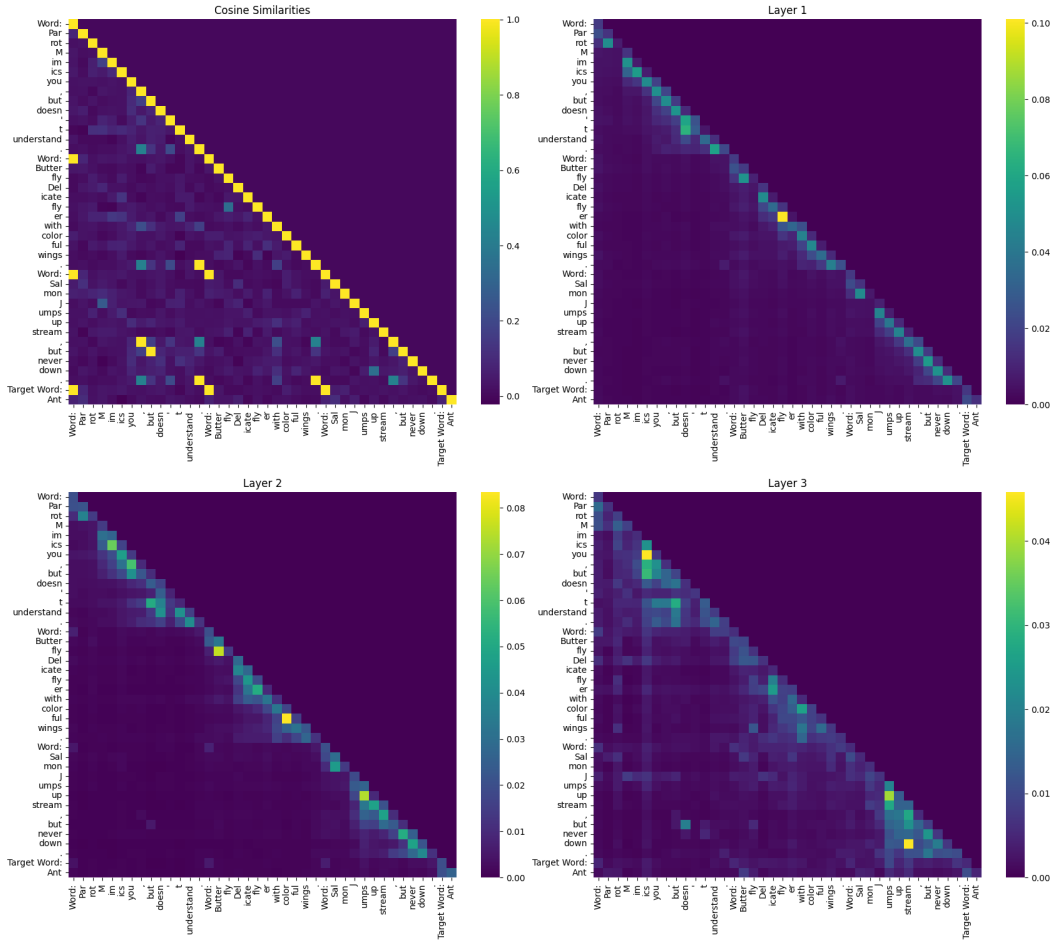


Fig. 8.6.: Embedding Cosine Similarities + Attention Maps for Target Word *"Ant"*, with $k = 3$ on Token Level for Layers $1, 2, 3$

that the newly generated embedding receives a broad distribution of probabilities, as previously discussed in "The Uniform Embedding Mixture Problem" in Section 7.2.3. The information is undoubtedly shared among all tokens but is widely balanced. Note that the sum of subwords' attention scores in the plots are added together for illustrative purposes, as the marginal attention scores lie in a small regime (see Figure 8.4), supporting our theoretical investigations of the distribution of log-normally distributed variables in Chapter 5.3.

8.3 Attention Maps Angular Information

In Chapter 6.3, we discussed the preservation of angular information between vectors, which serves as a precursor to pre-trained embeddings. Comparing

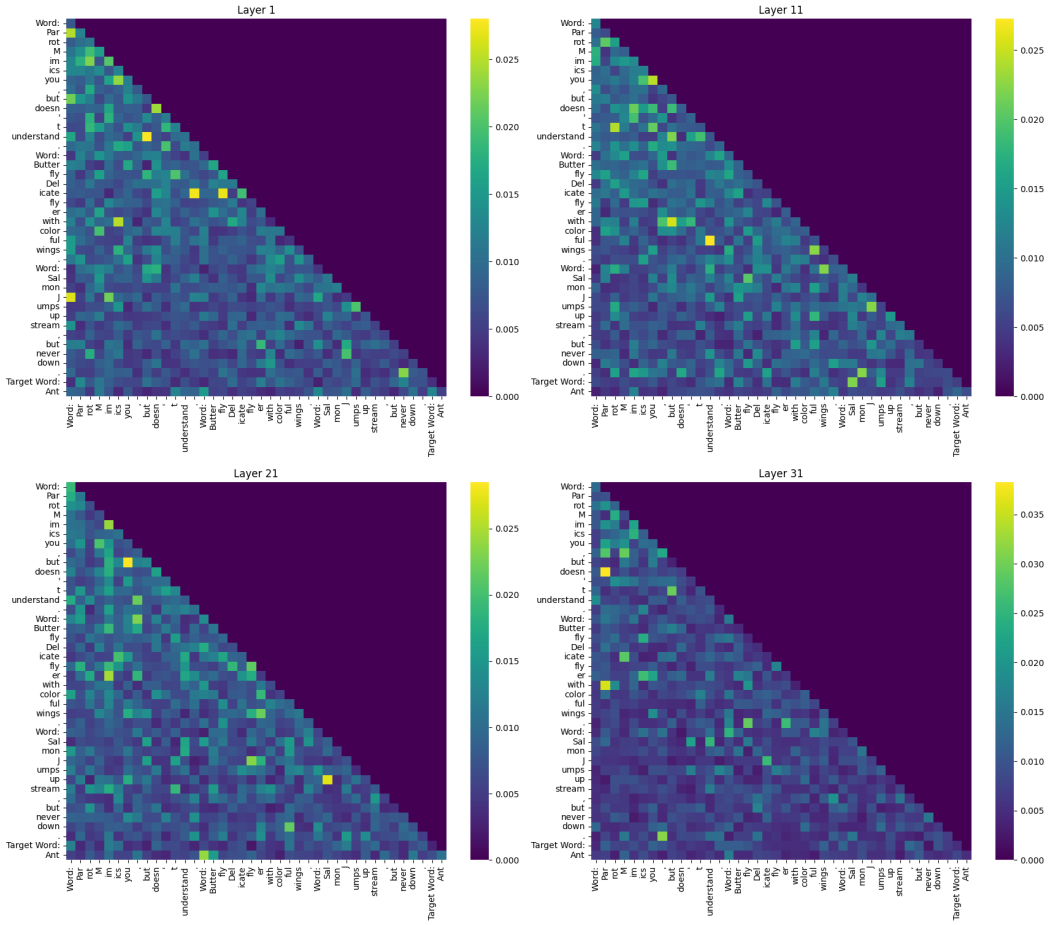


Fig. 8.7.: Attention Maps Randomized for *"Ant"*, $(k = 3)$, Generated Clue: *"calculation-sindi Arbitro hop shadow fallingSTrolligte convinced Qualitycott..."*

the normalized cosine similarity scores between tokens with the attention scores of chosen layers, I observed that the relationship between them is minimal; the model seems to dominate similarity matching in a more complex manner than anticipated. Figure 8.6 illustrates this point. This observation suggests that angular information can be misleading, as the attention mechanism is powerful enough to "overwrite" it.

8.4 Attention Maps Random Weight Matrices

Additionally, we discussed "The Uniform Embedding Mixture," so I decided to apply the Few-Shot prompt to a initial randomized Mistral 7B model (illustrated in plot 8.7). As we can see, the experiential investigation supports that

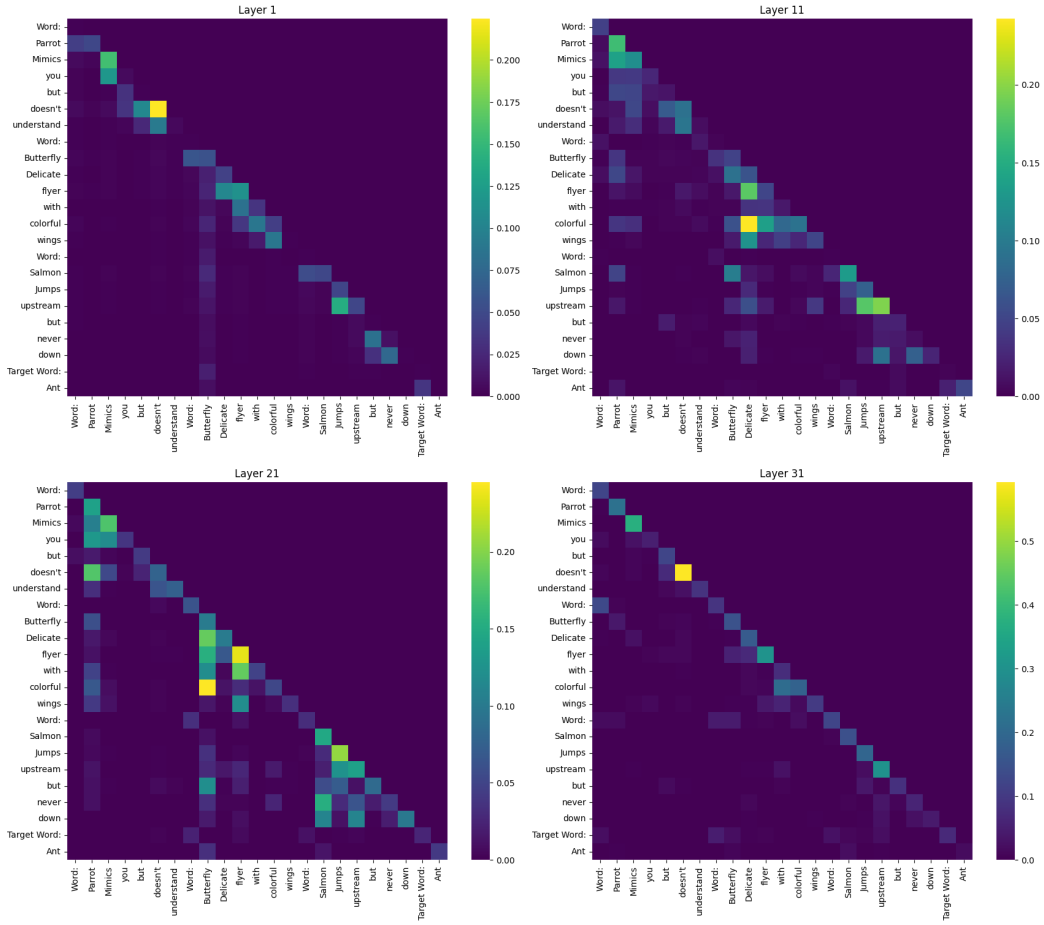


Fig. 8.8.: Attention Maps Word Level *"Ant"*, $(k = 3, \alpha = 2)$, Generated Clue: *"output a brief description for an answer:"*

supposition, even when RELU and Layer Normalization are in effect, as the attention scores approach a uniform distribution.

8.5 Attention Maps Alteration Singular Value Spectrum

Next, we manipulate the Singular Value Decomposition (SVD) of the attention weight matrices W_q and W_k, test it against our workflow, analyze the structure of the attention maps, and determine whether the model can still formulate coherent answers.

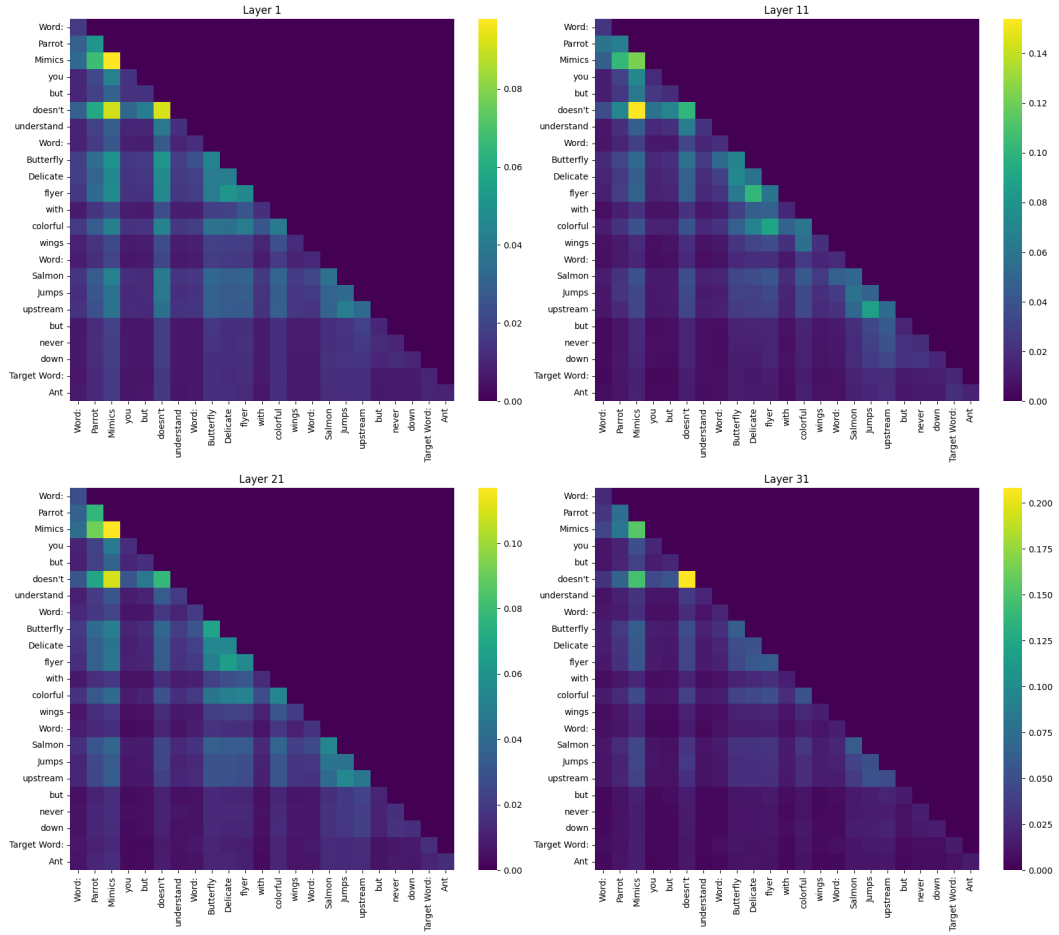


Fig. 8.9.: Attention Maps Word Level *"Ant"*, $(k = 3, \alpha = \frac{1}{2})$, Generated Clue: ""

To do this, we scale the magnitude of the singular value spectrum with a scaling variable $\alpha \in \mathbb{R}_{>0}$, equivalent to upscaling the radius of the hypersphere used in our analysis of log-normally distributed variables. This means multiplying the singular value spectrum of both W_q and W_k by α, thus scaling $\mathbb{E}[\langle W_q x, W_k y \rangle]$ by the factor α^2. This process is demonstrated for $\alpha = 2$ in Figure 8.8 and for $\alpha = \frac{1}{2}$ in Figure 8.9.

The attention patterns exhibit several trends discussed in the Softmax chapter, particularly the findings from Chapter 5.2.3. Upscaling causes the attention scores to become less balanced, converging to Dirac delta measures, resulting in tokens predominantly attending to themselves. Conversely, downscaling leads to flattening the probability distribution; this effect is not as strong as expected but still visible.

Interestingly, upscaling allows the model to generate coherent English sentences when examining the generated answers, though they don't fit the target word. This might be due to the over-proportional mixing of the marginal probability distributions. Interestingly, I found the model to often repeat the sentence of the passed prompt; e.g., in plot 8.8, the model generated *"output a brief description for an answer:"*, which was part of the original prompt context. Conversely, with downscaling, attention becomes pointless, and the attention scores converge to a float distribution, resulting in random sequences of Unicode characters without linguistic meaning.

Conclusion and Discussion

9

9.1 Conclusion and Discussion

This study aimed to investigate the attention mechanism within the Gaussian approximation framework, focusing on the data probability distribution and the characteristics of the model's weight matrices. Despite a predominantly theoretical approach, this investigation yields significant insights into the complexity of modeling the attention mechanism and the general inference dynamics of large language models. The research underscores the utility of advanced mathematical tools, particularly those from linear algebra, multivariate statistics, and information theory, in elucidating the intricacies of these models. In a nutshell, it became apparent why attention is so much more complicated than often assumed, as it expresses more than a simple measure of how important a particular token is for another, but instead, a tool to internally manage linear combinations of probability distributions.

Data Distribution under Gaussian Approximations. The Gaussian approximation has proven to be an insightful starting point, given the extensive theoretical groundwork that underpins it. The difficulty in mathematically condensing the statistics of arbitrary data distributions can be mitigated through linear joint Gaussian models, which interact with the model's nonlinear components. The pre-trained embedding of text data resides on a hypersphere and stores linguistic dependencies in their spherical positions and respective angular information. This allows for a more nuanced specification of the spectral properties of the marginal Gaussian distributions. Over time, the joint modeling framework facilitates generalization at both a micro-level, considering the marginal distribution of embedding vectors on the hypersphere, and a macro-level, using information theory tools like mutual information to model abstract linguistic modalities. Preserving linear combinations under linear maps is a significant advantage of Gaussians, enabling mathematical capture of interactions between Gaussians and linear components within the model, such as simple feedforward layers.

The Gaussian approximation feeds into the well-known attention formula $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, which aligns embedding vectors and applies SoftMax to generate a probability distribution that shapes new marginal distributions' linear combinations.

Statistics and Geometry of Softmax in Attention. The second major part of this study examined the SoftMax function from $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$. We established multiple connections between the input data's statistics (hypersphere model) and their exponentiated counterparts, which are log-normally distributed variables. For standard normal variables, corresponding to hyperspheres of radius $\frac{1}{n}$, we observed a flat value domain for the log-normal ratio. We demonstrated that linear combinations of vectors, $\sum_{i=1}^{n} \lambda_i v_i$, with $\sum_{i=1}^{n} \lambda_i = 1$, translate well into their exponentiated counterparts $\exp(\lambda_i v_i)$. This finding implies that linear contribution factors remain approximatively exhibited after exponentiation, with the low ratio value domain crucial for determining normalized attention scores and new linear combinations. Furthermore, we drew connections between Euclidean geometry and discrete probability spaces by contrasting the cosine or angle distance measure with cross-entropy for Softmax-transformed discrete probability spaces. The hypersphere model effectively demonstrates that the cross-entropy between two softmax-transformed vectors parametrizes their respective angle for orthogonal vectors, a common occurrence in randomized settings due to the Curse of Dimensionality.

Attention as Bilinear Form with Randomized and Learned Weight Matrices. In the third section, we examined the bilinear form $\langle W_q, W_k \rangle$ from $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$ to model the second uncertainty component, the matrix weight statistics. The singular value decomposition emerged as a useful tool for generalizing these interactions, aligning well with the joint Gaussian distributions of the embeddings. We analyzed the expected value and variance of the bilinear form, discovering the trace of the alignment matrix $W_q^T W_k$. This allowed us to understand how the weight matrix's base interacts with the data's Gaussians in untrained and trained states. Recognizing that learned weight matrices evolve from initially randomized matrices, we explored how the trace of independent randomized matrices follows a chi-squared distribution, approximating a normal distribution. This understanding deepens our grasp of the scaling effects of $\langle W_q x, W_k y \rangle$. We also investigated how the singular value decomposition aids in comprehending diagonalized systems, the challenges therein, and the impact of altering the singular value spectrum on $\text{Tr}(W_q^T W_k)$. The pre-trained angular information $\langle x, y \rangle$ is preserved in the

calculated variance, referencing previously utilized hyperspheres. Random matrix theory thus provides insight into the initial configuration's role in model training and how trained matrices may alter this behavior.

Gaussians Mixture Models and the "Uniform Embedding Mixture Problem". The fourth and final theoretical part addressed the mixture of marginal probability components via the value weight matrix W_v in Softmax $\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, synthesizing the individual parts of the attention mechanism. Creating Gaussian mixture models through linear combinations, which are subsequently processed by the next attention mechanism operation, reveals critical aspects of the model's function. The flatness of the log-normal distribution, combined with the absence of coupling in the base alignment matrix, highlights the "Uniform Embedding Mixture Problem," raising concerns about the transformer's ability to distinguish individual embedding components. As suggested by Anthropic's research, skip connections with consecutive layer normalization and the non-linear RELU function play crucial roles in addressing this issue. This study emphasizes the importance of understanding both additive and multiplicative components to fully grasp why Transformer models exhibit such remarkable performance, increasingly dominating various applications in human life.

Experiments: Attention in Few-Shot Prompting. This experimental journey investigated the structure of attention patterns in a current NLP setting: Few-Shot Prompting, where the model generates a new clue to a word of a category in the same style based on previous shots/examples. The attention patterns provide insights into how the model reasons about these prompts on a broad scale. We observed the same attention structure progression described in current literature: initial diagonal dominance, followed by dispersion and a subsequent contraction of the attention values. Moreover, the angular information from the pre-trained embeddings is slightly preserved, and random weight matrices converge to a flattened attention map. Additionally, we noted that the value domain of attention scores is often low. This becomes even more apparent when scaling the singular value matrices of the bilinear form. We found that models can still formulate valid sentences with slight upscaling despite partly repeating the input. However, downscaling the singular values prevents the model from operating, resulting in incoherent token chains.

9.2 Limitations

This study faces several limitations in both its theoretical and practical components.

Data Distribution under Gaussian Approximations. Foremost, approximating complex data distributions with their stochastic moments in the form of Gaussians presents significant challenges. Estimating the accuracy of these approximations is difficult due to the arbitrary nature of the dataset probability distribution. Consequently, all subsequent conclusions and numerical approximations should be approached with caution, as time constraints prevented a comprehensive numerical development of the argumentative chain. Incorporating geometrical rotational information from pre-trained embeddings in NLP provides some justification for these assumptions, though a deeper examination of singular value distributions would offer more insights. However, in practical language generation tasks, the vast number of potential subsequent words diminishes the significance of approximating linear combinations, even with Bayesian conditioning. The symmetry of the marginal Gaussians is shaped by the drawback that the non-symmetric RELU operation with subsequent Layer Normalization results in probability distributions different from Gaussians, raising concerns about the robustness of this modeling approach.

Statistics and Geometry of Softmax in Attention. Another limitation lies with the SoftMax function. My primary goal was to reconcile linear combinations in Euclidean geometry with probabilistic discrete function spaces. The use of information theory is justified by the differentiable contrastive measure during differentiation, yet the logarithm introduces numerous scaling effects. The combination of exponentiation, normalization, and logarithm in information theory creates a substantial mathematical workload, complicating the precise determination of higher moments. My numerical approximations to embed the second moment (variance) into Gaussians for estimating the product of two asymmetrically distributed random variables might seem presumptuous. However, as long as the numerics support these approximations, they remain a valid foundation. Translating angular information as a similarity/distance measure into the KL divergence of probability distance on a geometrically preserving hypersphere manifold under SLERP interpolation proved too complex, though it remains a potential avenue for future exploration.

Attention as Bilinear Form with Randomized and Learned Weight Matrices. In the third section, the attempt to connect the asymmetric bilinear form $\langle W_Q x, W_K y \rangle$ with the singular value decomposition of W_Q and W_K, considering scaling disparities between randomized initial matrices and learned matrices, presents another challenge. Ignoring biases in this equation for theoretical findings is problematic, as the structure resembles a multivariate polynomial with non-quadratic mixed terms intertwined by both W_Q and W_K. Even if both matrix kernels are zero, this involves many higher moments. Similar to the data issue, we need to set scaling indices like $tr(W_Q^T W_K)$, whose evolution we cannot precisely derive, as they are closely related to the overall complexity of training dynamics. Thus, post-processing angular information from intermediate results complicates the use of angular information as a straightforward explanation for creating attention values. In simpler terms, the formation of attention patterns in the second theoretical part cannot be solely explained by contrasting relative geometry between intermediate embedding representations due to the many degrees of freedom in the asymmetry of the bilinear form, shaping geometry in complex ways.

Gaussians Mixture Models and the "Uniform Embedding Mixture Problem". This complexity leads to formulating the "Uniform Embedding Mixture Problem." Each component is challenging in itself, making their reasonable and comprehensive integration difficult, underscoring the arduous nature of understanding the power of transformer models.

Experiments: Attention in Few-Shot Prompting. From the practical perspective, I discussed much in the respective section. Still, it is definitely worth noting that when expecting intermediate results like Gaussians, all advanced components like RELU, Batch Normalization, and Skip Connections impact the sampled results. Marginalizing the object of study out of this accumulation requires a huge effort and better and more advanced modeling, which also extends to those components I have not reached in the scope of this study. The work has not already considered the joint Gaussians of the data distribution. However, modeling Gaussians with a thousand components undoubtedly requires many computing resources, not to mention the numerical issues encountered when working on small singular-value spectra.

9.3 Future Work

Future work is difficult to outline concretely due to the theory-heavy nature of the individual components involved. However, strengthening the four main chapters—joint Gaussian, SoftMax, bilinear form, and Gaussian Mixture models—is crucial.

One aspect to explore is how the statistics of the bilinear form interact with log-normal distributed variables. The geometry of probability spaces of SoftMax-transformed vectors should also be further investigated. While this study delved into SLERP interpolation to match Euclidean distance and cross-entropy distance, other information theoretical measures should be connected to geometry, such as entropy, Jensen distance, and mutual information.

The joint Gaussian approach connects KL divergence and mutual information to linear algebra. It would be insightful to analyze their behavior in new marginal Gaussian Mixture models after an attention pass, particularly considering the statistics of attention scores. With that, I also considered classifying distributions of singular values more concretely in terms of a geometric meaning, like considering a vector on a sphere with magnitude and then calculating the properties we inspected throughout this work with this new representation.

One interesting aspect revolves around how the Transformer model organizes these Mixture components and how information exchange occurs. I briefly touched on this when discussing Mutual Information on Gaussian Mixture models, but it has yet to be explored further.

Finally, further study of the proposed "Uniform Mixture Problem" is necessary. Tackling the question of why attention patterns exhibit particular behaviors more mathematically is an intriguing and complex challenge that extends beyond the scope of this study. Understanding how information-theoretic measures concerning the Mutual Information for the Gaussian Mixture model promises insightful results, potentially with the combination of the low-rank property of the attention operator.

Bibliography

[1]Joshua Ainslie, James Lee-Thorp, Michiel de Jong, et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*. 2023. arXiv: 2305.13245 [cs.CL] (cit. on p. 75).

[2]Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML] (cit. on pp. 13, 23).

[3]Bing Bai, Jian Liang, Guanhua Zhang, et al. *Why Attentions May Not Be Interpretable?* 2021. arXiv: 2006.05656 [stat.ML] (cit. on p. 22).

[4]Srinadh Bhojanapalli, Ayan Chakrabarti, Himanshu Jain, et al. *Eigen Analysis of Self-Attention and its Reconstruction from Partial Computation*. 2021. arXiv: 2106.08823 [cs.LG] (cit. on p. 24).

[5]Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. *Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers*. 2016. arXiv: 1604.00825 [cs.CV] (cit. on p. 22).

[6]Hila Chefer, Shir Gur, and Lior Wolf. *Transformer Interpretability Beyond Attention Visualization*. 2021. arXiv: 2012.09838 [cs.CV] (cit. on p. 22).

[7]Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, et al. *A Simple and Effective Positional Encoding for Transformers*. 2021. arXiv: 2104.08698 [cs.CL] (cit. on p. 11).

[8]Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. *What Does BERT Look At? An Analysis of BERT's Attention*. 2019. arXiv: 1906.04341 [cs.CL] (cit. on pp. 2, 21, 82).

[9]Amit Daniely, Roy Frostig, and Yoram Singer. *Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity*. 2017. arXiv: 1602.05897 (cit. on p. 69).

[10]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL] (cit. on p. 77).

[11]Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: (2021). arXiv: 2010.11929 [cs.CV] (cit. on p. 23).

[12]Nelson Elhage, Tristan Hume, Catherine Olsson, et al. "Toy Models of Superposition". In: *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/toy-model/index.html (cit. on pp. 25, 74).

[13]Nelson Elhage, Neel Nanda, Catherine Olsson, et al. “A Mathematical Framework for Transformer Circuits”. In: *Transformer Circuits Thread* (2021). https://transformer-circuits.pub/2021/framework/index.html (cit. on pp. 25, 74).

[14]Ki Wai Fong and Shingyu Leung. *Spherical Essentially Non-Oscillatory (SENO) Interpolation*. 2022. arXiv: 2212.01963 (cit. on p. 40).

[15]Jonathan Frankle and Michael Carbin. *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks*. 2019. arXiv: 1803.03635 (cit. on p. 22).

[16]Maosheng Guo, Yu Zhang, and Ting Liu. “Gaussian Transformer: A Lightweight Approach for Natural Language Inference”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 6489–6496 (cit. on p. 23).

[17]Marwane Ben Hcine and Ridha Bouallegue. *On the Approximation of the Sum of Lognormals by a Log Skew Normal Distribution*. 2015. arXiv: 1502.03619 (cit. on p. 36).

[18]Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL] (cit. on pp. 13, 75).

[20]Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. *Revealing the Dark Secrets of BERT*. 2019. arXiv: 1908.08593 [cs.CL] (cit. on pp. 21, 82).

[21]Siddharth Krishna Kumar. *On weight initialization in deep neural networks*. 2017. arXiv: 1704.08863 (cit. on pp. 18, 50).

[22]Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. *A Survey of Transformers*. 2021. arXiv: 2106.04554 [cs.LG] (cit. on p. 21).

[23]Yang Liu, Yao Zhang, Yixin Wang, et al. “A Survey of Visual Transformers”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023), pp. 1–21 (cit. on p. 23).

[24]Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. *Introduction to Random Matrices*. Springer International Publishing, 2018 (cit. on pp. 18, 53).

[25]Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI] (cit. on p. 22).

[26]Xiuqing Lv, P. Zhang, Sunzhu Li, Guobing Gan, and Yueheng Sun. “LightFormer: Light-weight Transformer Using SVD-based Weight Transfer and Parameter Sharing”. In: *Annual Meeting of the Association for Computational Linguistics*. 2023 (cit. on p. 24).

[27]Paul Michel, Omer Levy, and Graham Neubig. *Are Sixteen Heads Really Better than One?* 2019. arXiv: 1905.10650 [cs.CL] (cit. on pp. 21, 82).

[28]Tanmoy Mukherjee and Timothy Hospedales. “Gaussian Visual-Linguistic Embedding for Zero-Shot Recognition”. English. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016 ; Conference date: 01-11-2016 Through 05-11-2016. Association for Computational Linguistics (ACL), Nov. 2016, pp. 912–918 (cit. on p. 23).

[29]Lakshmi Narayan Pandey, Rahul Vashisht, and Harish G. Ramaswamy. “On the Interpretability of Attention Networks”. In: *Proceedings of The 14th Asian Conference on Machine Learning*. Ed. by Emtiyaz Khan and Mehmet Gonen. Vol. 189. Proceedings of Machine Learning Research. PMLR, Dec. 2023, pp. 832–847 (cit. on p. 22).

[30]Dipanjyoti Paul, Arpita Chowdhury, Xinqi Xiong, et al. *A Simple Interpretable Transformer for Fine-Grained Image Classification and Analysis*. 2024. arXiv: 2311.04157 [cs.CV] (cit. on p. 23).

[31]Jeffrey Pennington, Richard Socher, and Christopher Manning. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543 (cit. on pp. 2, 11).

[32]Bosheng Qin, Juncheng Li, Siliang Tang, and Yueting Zhuang. *DBA: Efficient Transformer with Dynamic Bilinear Low-Rank Attention*. 2022. arXiv: 2211.16368 (cit. on pp. 24, 70).

[33]Colin Raffel, Noam Shazeer, Adam Roberts, et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 (cit. on p. 77).

[35]Waddah Saeed and Christian Omlin. “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities”. In: *Knowledge-Based Systems* 263 (2023), p. 110273 (cit. on p. 22).

[36]Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. “Representational Strengths and Limitations of Transformers”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, et al. Vol. 36. Curran Associates, Inc., 2023, pp. 36677–36707 (cit. on p. 24).

[37]Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359 (cit. on p. 22).

[38]A. M. Sengupta and P. P. Mitra. *Distributions of Singular Values for Some Random Matrices*. 1997. arXiv: cond-mat/9709283 [cond-mat.stat-mech] (cit. on pp. 18, 53).

[39]Sofia Serrano and Noah A. Smith. *Is Attention Interpretable?* 2019. arXiv: 1906.03731 [cs.CL] (cit. on pp. 2, 22).

[40]Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL] (cit. on pp. 13, 76).

[41]Hugo Touvron, Louis Martin, Kevin Stone, et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL] (cit. on pp. 13, 75).

[42]Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. *Transformer Dissection: A Unified Understanding of Transformer's Attention via the Lens of Kernel*. 2019. arXiv: 1908.11775 [cs.LG] (cit. on p. 24).

[43]Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL] (cit. on pp. 2, 5, 10, 12, 14, 21, 76).

[44]Luke Vilnis and Andrew McCallum. *Word Representations via Gaussian Embedding*. 2015. arXiv: 1412.6623 [cs.CL] (cit. on p. 23).

[45]Alex Wang, Amanpreet Singh, Julian Michael, et al. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. 2019. arXiv: 1804.07461 [cs.CL] (cit. on p. 22).

[46]Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. *Linformer: Self-Attention with Linear Complexity*. 2020. arXiv: 2006.04768 [cs.LG] (cit. on pp. 12, 24, 25, 70).

[47]Hanqi Yan, Lin Gui, Wenjie Li, and Yulan He. "Addressing token uniformity in transformers via singular value transformation". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, Jan. 2022, pp. 2181–2191 (cit. on pp. 23, 24, 72).

[48]Sumu Zhao, Damian Pascual, Gino Brunner, and Roger Wattenhofer. *Of Non-Linearity and Commutativity in BERT*. 2021. arXiv: 2101.04547 [cs.CL] (cit. on p. 23).

[49]Daquan Zhou, Bingyi Kang, Xiaojie Jin, et al. *DeepViT: Towards Deeper Vision Transformer*. 2021. arXiv: 2103.11886 [cs.CV] (cit. on p. 23).

Webpages

[@19]Richard Kleeman. *Information Theory and Predictability Lecture 7: Gaussian Case*. 2017. URL: https://math.nyu.edu/~kleeman/infolect7.pdf (visited on June 25, 2024) (cit. on p. 8).

[@34]Sebastian Raschka. *Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch*. 2024. URL: https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html (visited on June 19, 2024) (cit. on p. 15).

List of Figures

List of Tables

Appendix Additional Results

A

A.1 Few-Shot Prompting Generated Clues Results

Word	Generated Clue
Ant	Small, yet mighty workers in a complex society.
Bee	Honey-producing insect.
Fox	Cunning, red-hued forest dweller.
Raven	Large, black bird with a complex call.
Tarantula	Large, hairy, venomous arachnid.
Lemming	Small, rodent-like Arctic dweller.
Mink	Furred, semi-aquatic mammal.
Dog	Man's best friend.
Hare	Famous relative of the rabbit.
Hippopotamus	Large, semi-aquatic mammal with massive jaws.
Koi	Ornamental fish prized for its vibrant scales.
Whelp	Young wild animal, often a canid.
Shrew	Small, nocturnal mammal with a long snout.
Chinchilla	Furry rodent with distinctive quills.
Centipede	Long, many-legged invertebrate.
Cockroach	Nocturnal, six-legged pest.
Beetle	Six-legged, hard-shelled insect.
Toucan	Large, colorful bird with a distinctive, oversized beak.
Dormouse	Small, nocturnal rodent with large, baggy cheeks.
Mantaray	Large, flat, stingray relative with wing-like pectoral fins.
Wolverine	Ferocious, weasel-like creature with retractable claws.
Grizzlybear	Large, brown bear with a hump.
Earthworm	Organism that aids soil health.
Swallow	Flying bird with a forked tail and aerial acrobatics.
Honeybadger	Rascal with a sweet tooth and a fearless attitude.
Mandrill	Large, colorful primate of Africa.
Ayeaye	Malagasy lemur with a long, thin tail and a distinctive call.
Bison	Large, shaggy herbivore.
Kea	Intelligent, colorful parrot of New Zealand.
Axolotl	Aquatic salamander, not fully grown.

Tab. A.1.: Generated clues (3 shots) for target words of the category "Animals"

Word	Generated Clue
Ant	Small, social insect with a big heart.
Bee	Honey-producing, pollen-collecting insect.
Fox	Cunning, red-tailed forest dweller.
Raven	Black, intelligent bird.
Tarantula	Large, hairy, venomous spider.
Lemming	Small, rodent-like Arctic dweller.
Mink	Furred, semi-aquatic mammal.
Dog	Man's best friend.
Hare	Faster than a tortoise, but not a rabbit.
Hippopotamus	Large, semi-aquatic mammal.
Koi	Ornamental, colorful fish.
Whelp	Young wild animal.
Shrew	Small, active, and long-snouted rodent.
Chinchilla	Soft, furry rodent with quill-like hairs.
Centipede	Long, many-legged arthropod.
Cockroach	Nocturnal, common household pest.
Beetle	Explorer of decay, six-legged.
Toucan	Large, colorful, beak-featured bird.
Dormouse	Small, hibernating rodent.
Mantaray	Large, flat, stingray relative.
Wolverine	Ferocious, weasel-like predator.
Grizzlybear	Great Bear, powerful, brown predator.
Earthworm	Organic soil dweller.
Swallow	Flying bird with aerial acrobatics.
Honeybadger	Rascal with a stripe, fearless and badgered.
Mandrill	Large, colorful primate of Africa.
Ayeaye	Malagasy primate with a long, thin tail.
Bison	Large, shaggy herbivore.
Kea	Intelligent, colorful parrot of New Zealand.
Axolotl	Aquatic salamander with external gills.

Tab. A.2.: Generated clues (5 shots) for target words of the category "Animals"

Word	Generated Clue
Ant	Small, social insect.
Bee	Fuzzy pollinator.
Fox	Cunning, red-coated woodland creature.
Raven	Black, intelligent bird.
Tarantula	Large, hairy, venomous spider.
Lemming	Small, rodent-like Arctic dweller.
Mink	Small, agile, semi-aquatic mammal.
Dog	Man's best friend.
Hare	Faster than a tortoise, but not a rabbit.
Hippopotamus	Large, semi-aquatic mammal.
Koi	Ornamental, colorful fish.
Whelp	Young wolf or canid.
Shrew	Small, active rodent.
Chinchilla	Furry rodent with quill-covered fur.
Centipede	Many-legged, swift ground dweller.
Cockroach	Nocturnal, six-legged pest.
Beetle	Exoskeleton-wearing, six-legged insect.
Toucan	Colorful, large-beaked bird.
Dormouse	Small, hibernating rodent.
Mantaray	Flat, ray-like fish with triangular pectoral fins.
Wolverine	Ferocious, weasel-like creature.
Grizzlybear	Large, brown bear with a hump.
Earthworm	Invertebrate that aids soil.
Swallow	Aerial, aerial-feeding bird.
Honeybadger	Ferocious, honey-loving marsupial.
Mandrill	Large, colorful primate.
Ayeaye	Lemur with a long, thin middle finger.
Bison	Large, shaggy, North American mammal.
Kea	New Zealand's mountain parrot.
Axolotl	Aquatic, salamander with external gills.

Tab. A.3.: Generated clues (8 shots) for target words of the category "Animals"

A.2 Complete Few-Shot Prompting Attention Map pass over all Attention Layers

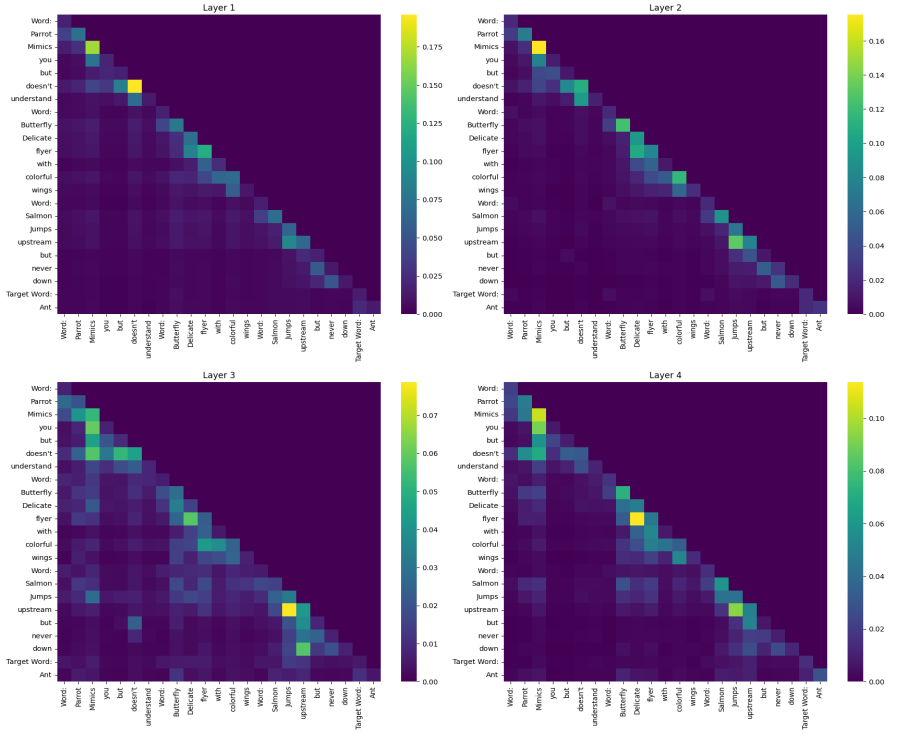


Fig. A.1.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 1-4

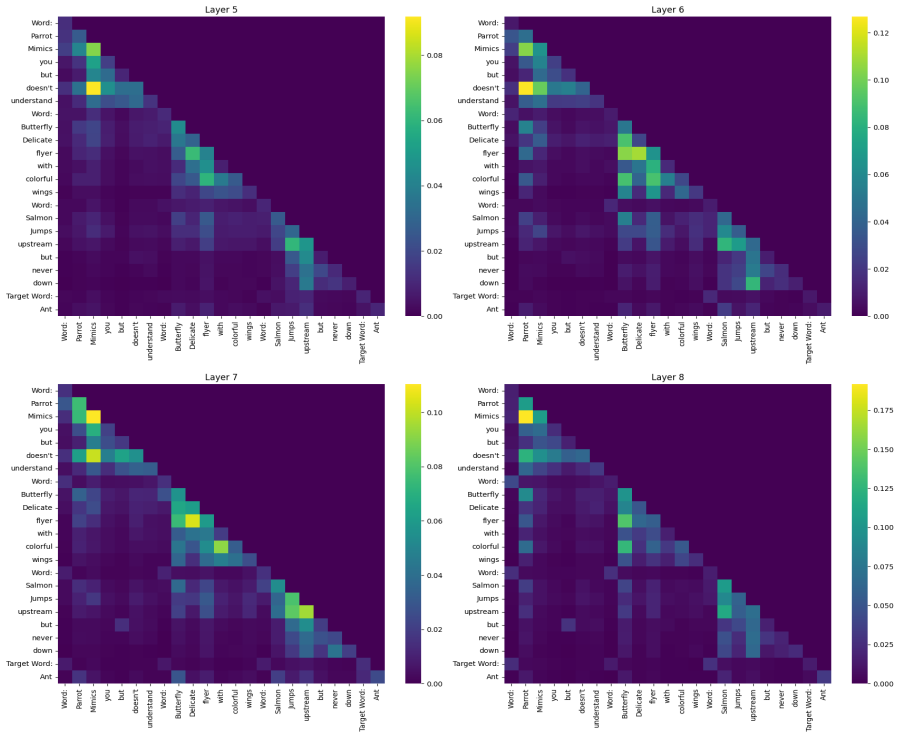


Fig. A.2.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 5-8

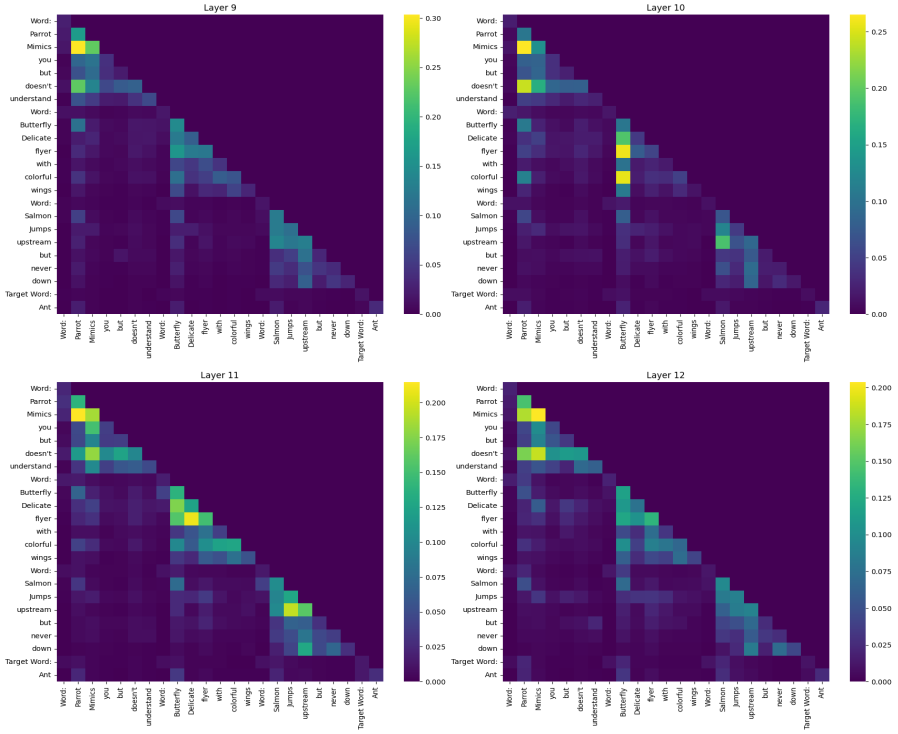


Fig. A.3.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 9-12

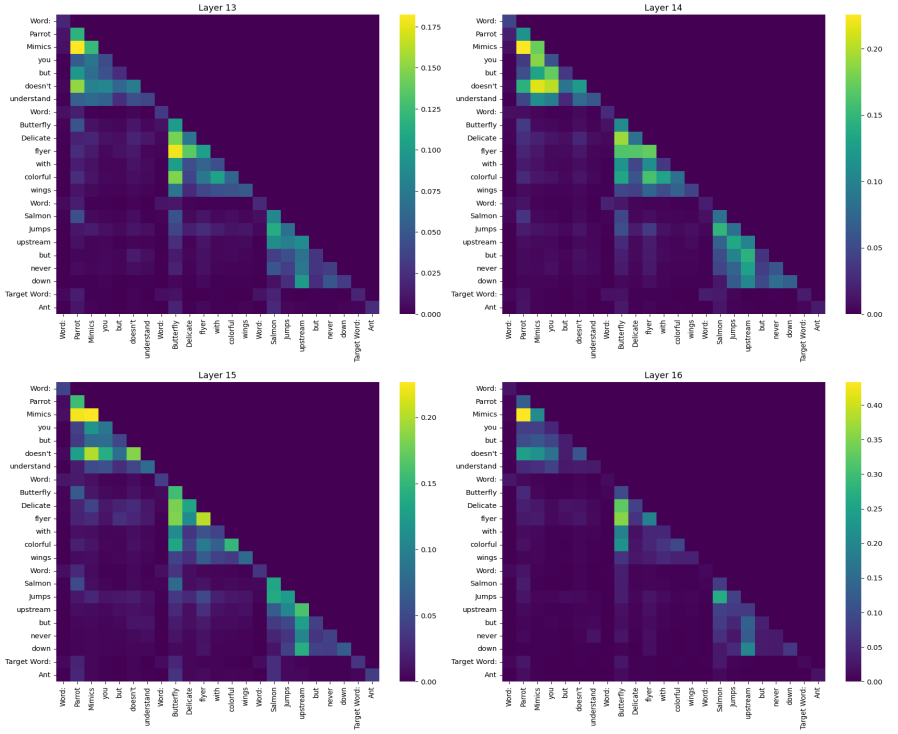


Fig. A.4.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 13-16

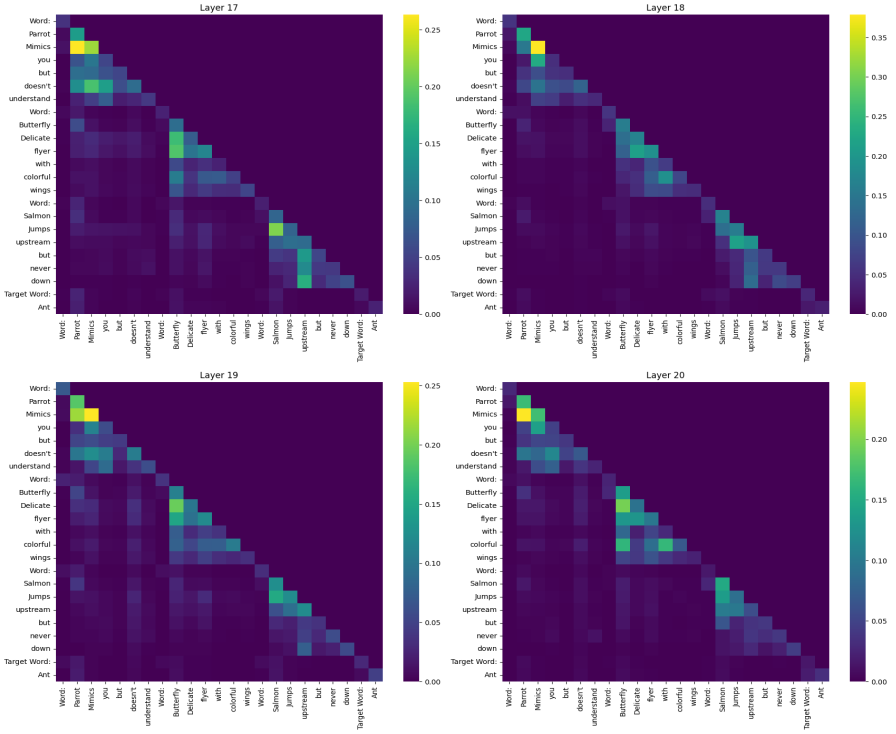


Fig. A.5.: Attention Maps *"Ant"* ($k = 3$), Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 17-20

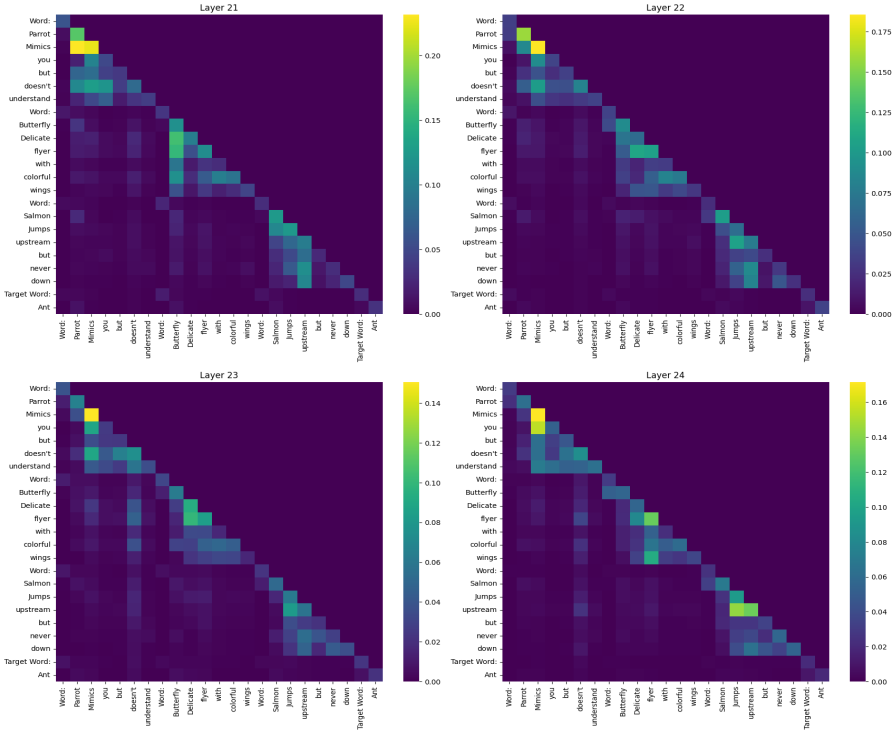


Fig. A.6.: Attention Maps *"Ant"* ($k = 3$), Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 21-24

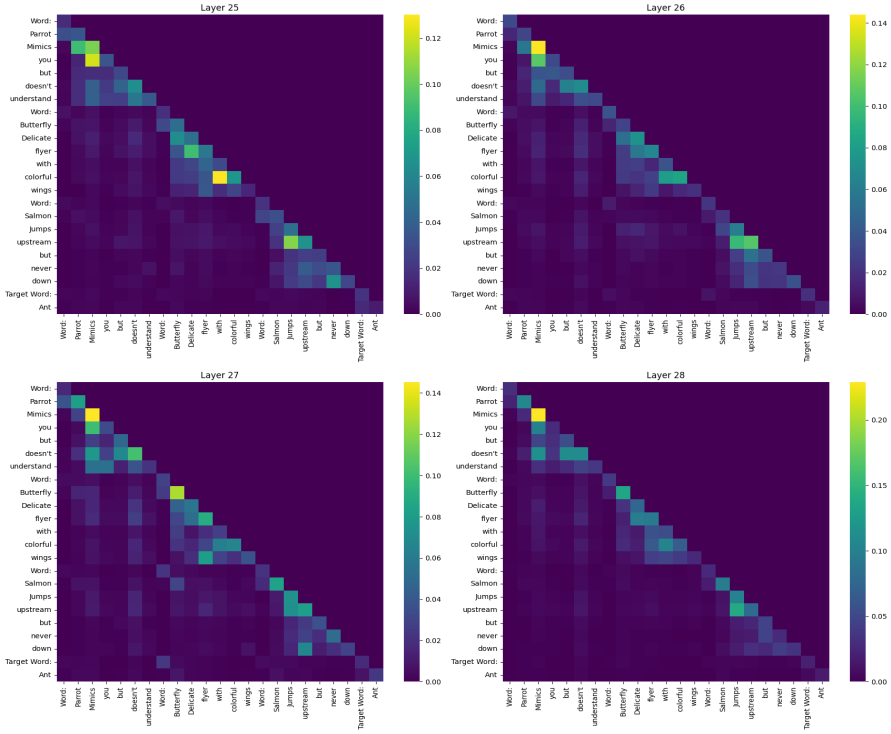


Fig. A.7.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 25-28

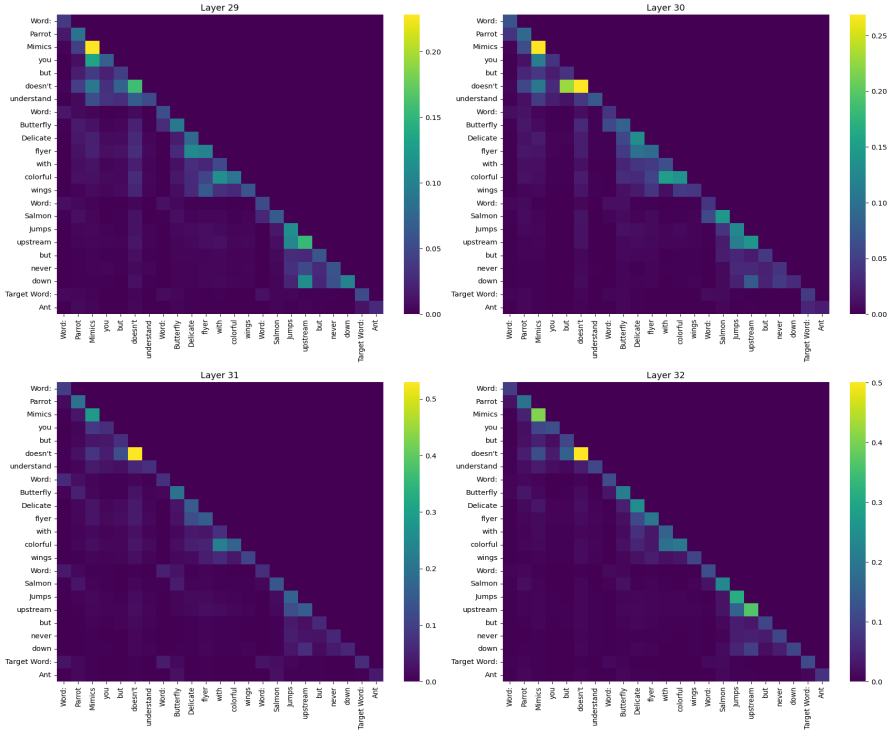


Fig. A.8.: Attention Maps *"Ant"* $(k = 3)$, Generated Clue: *"Small, yet mighty workers in a complex society."*, *Layers* 29-32

A.3 Attention Maps for other generated clues

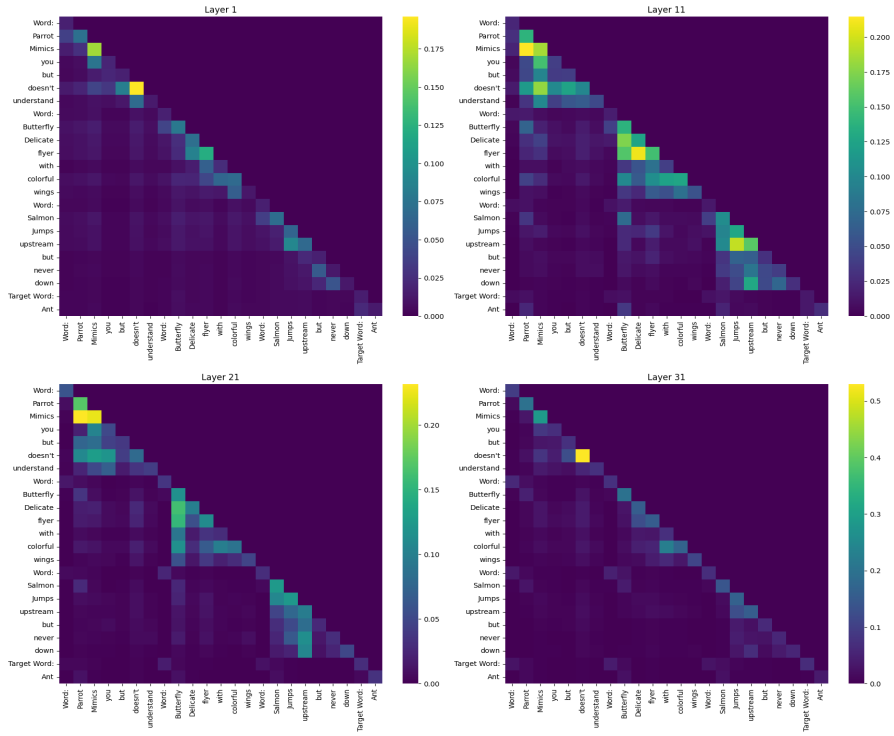


Fig. A.9.: Attention Maps *"Ant"* ($k = 3$), Generated Clue: *"Small, yet mighty workers in a complex society."*

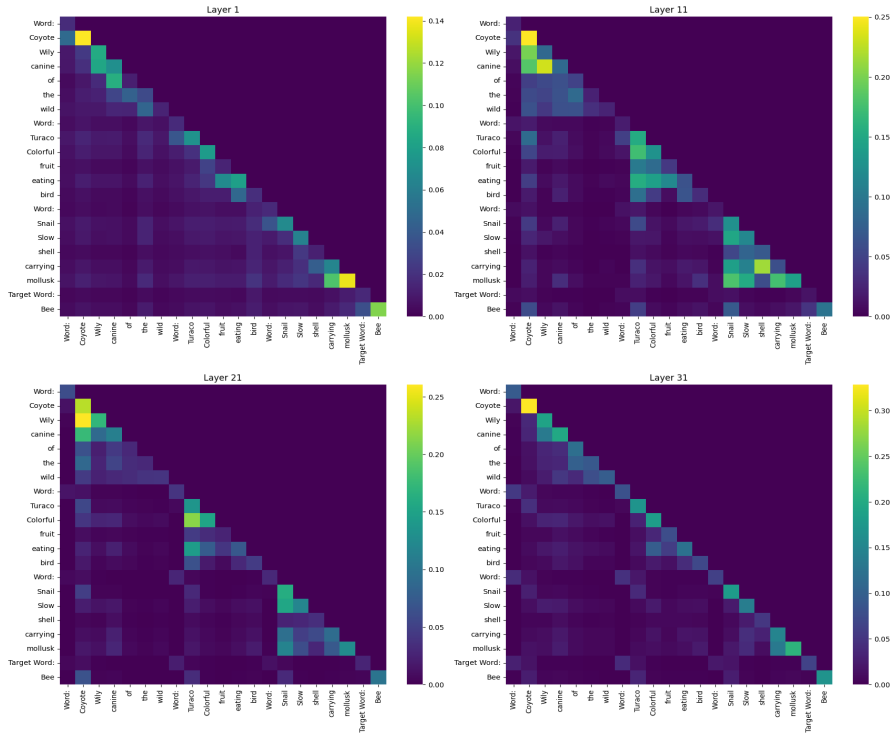


Fig. A.10.: Attention Maps *"Bee"* ($k = 3$), Generated Clue: *"Honey-producing insect."*

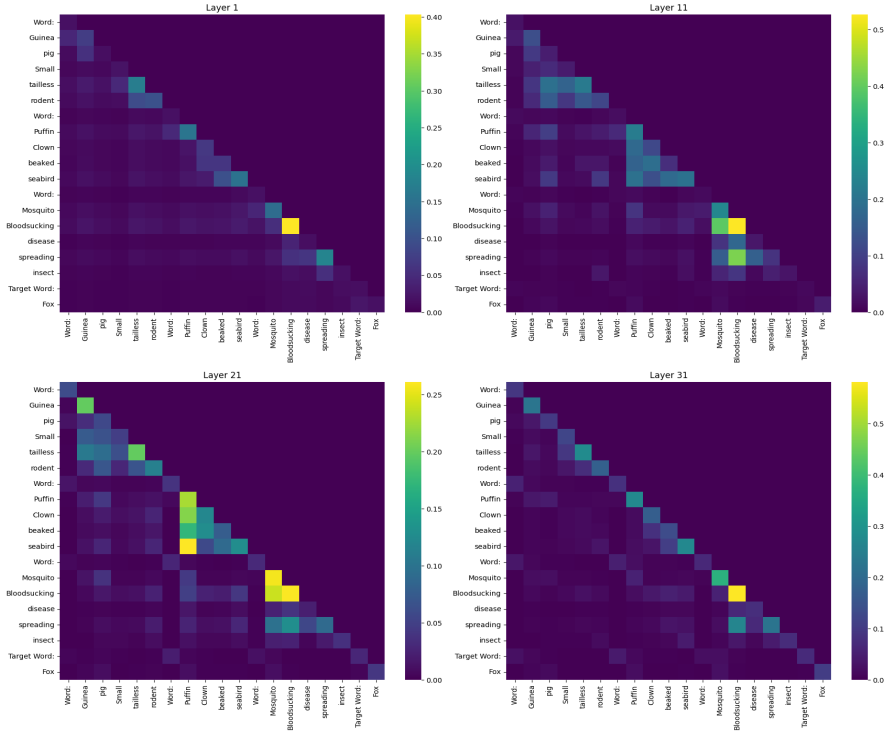


Fig. A.11.: Attention Maps *"Fox"* $(k = 3)$, Generated Clue: *"Cunning, red-hued forest dweller."*

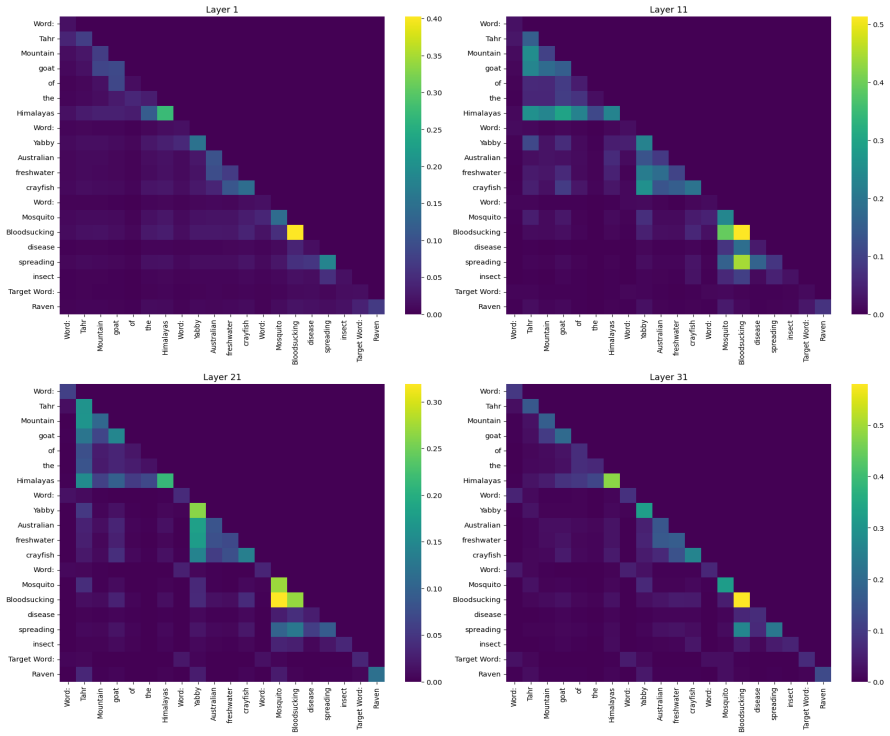


Fig. A.12.: Attention Maps *"Raven"* $(k = 3)$, Generated Clue: *"Large, black bird with a complex call."*

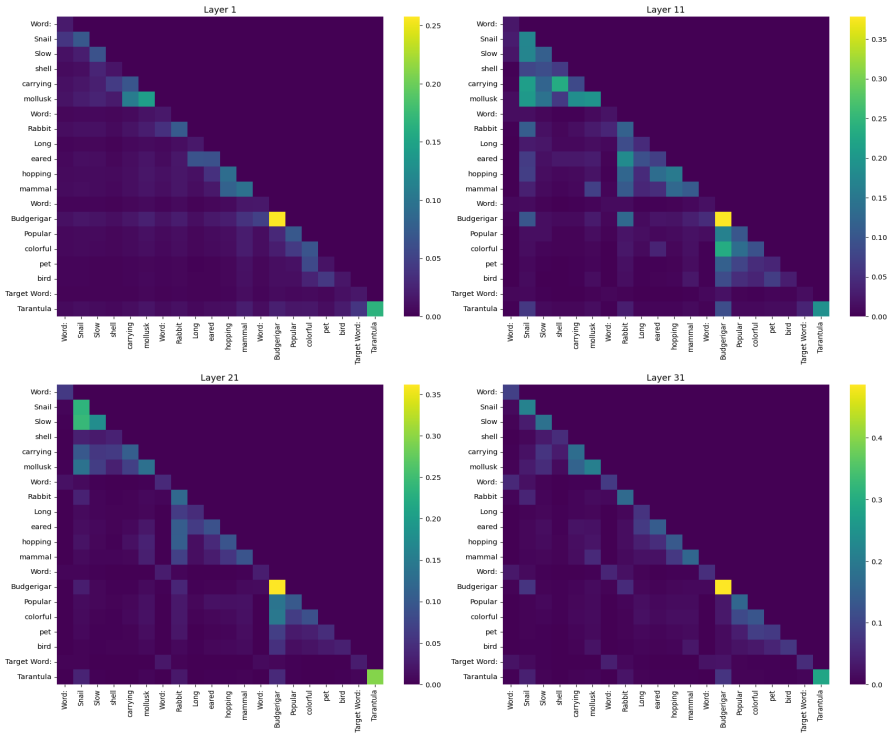


Fig. A.13.: Attention Maps *"Tarantula"* $(k = 3)$, Generated Clue: *"Large, hairy, venomous arachnid."*

A.4 Attention Maps for different numbers of previous example shots

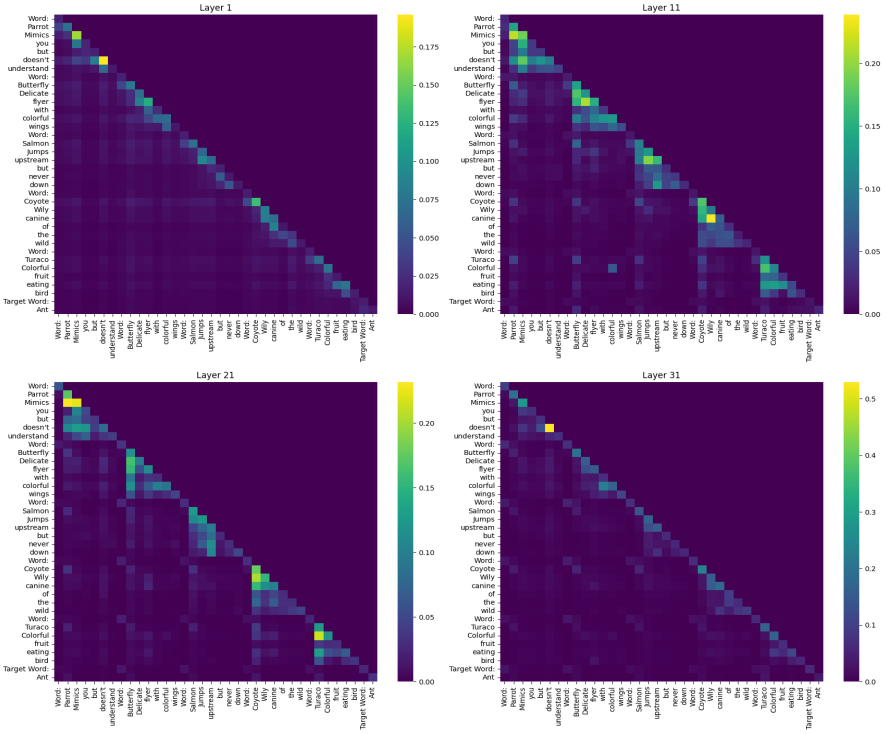


Fig. A.14.: Attention Maps *"Ant"* $(k = 5)$, Generated Clue: *"Small, social insect with a big heart."*

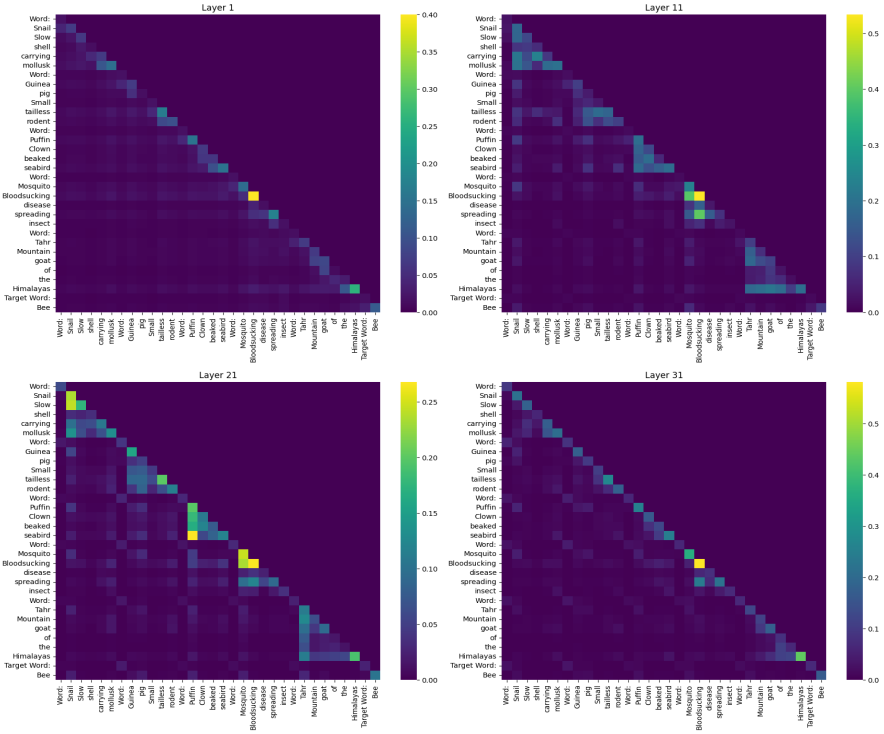


Fig. A.15.: Attention Maps *"Bee"* $(k = 5)$, Generated Clue: *"Honey-producing, pollen-collecting insect."*

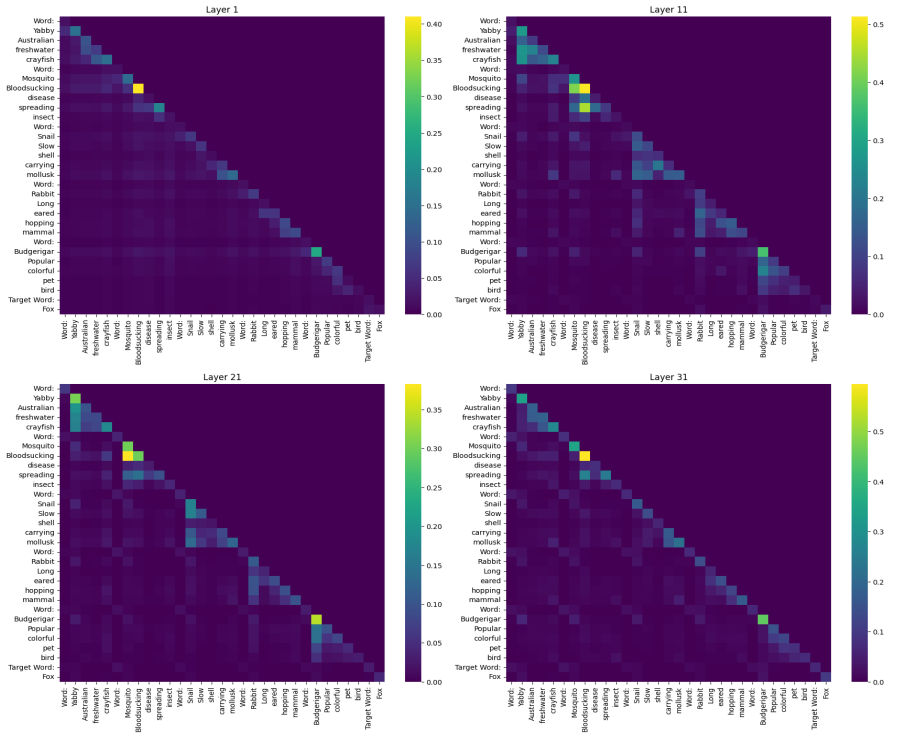


Fig. A.16.: Attention Maps *"Fox"* ($k = 5$), Generated Clue: *"Cunning, red-tailed forest dweller."*

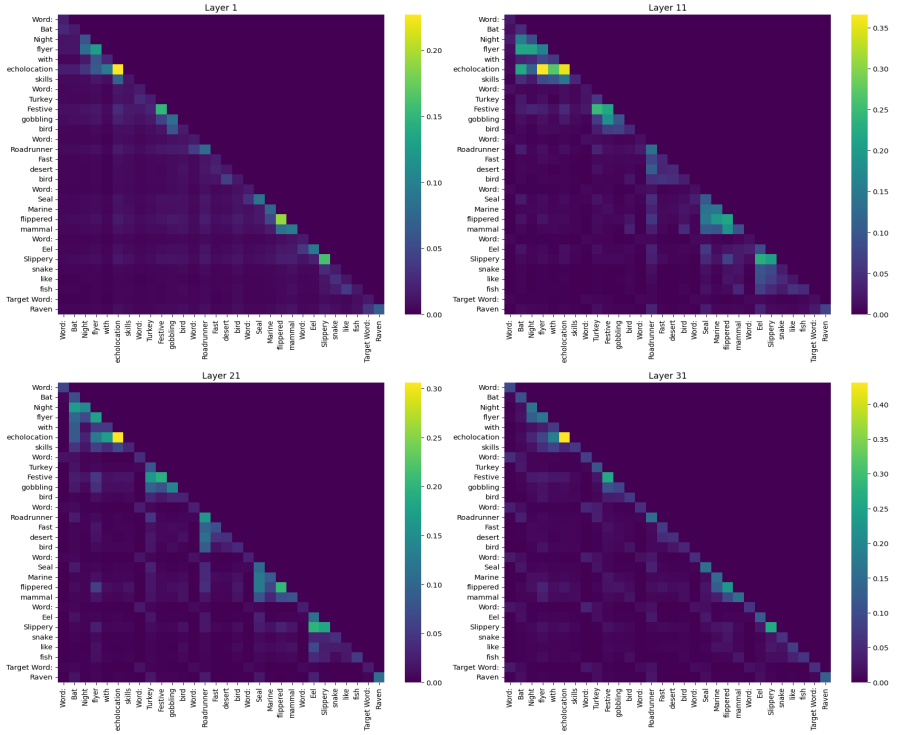


Fig. A.17.: Attention Maps *"Raven"* ($k = 5$), Generated Clue: *"Black, intelligent bird."*

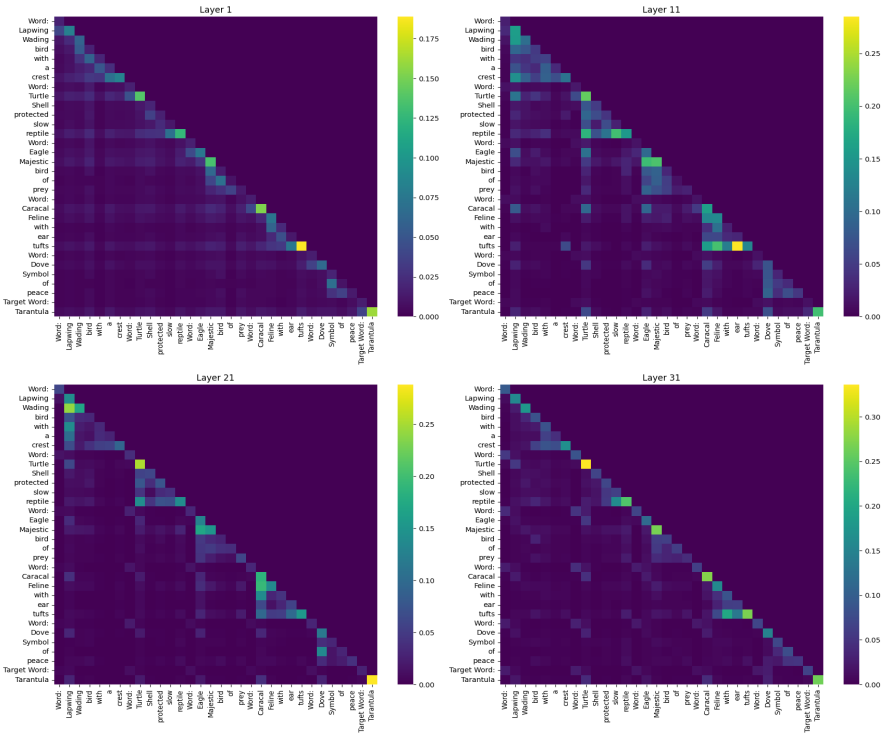
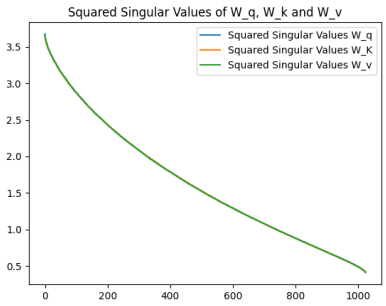
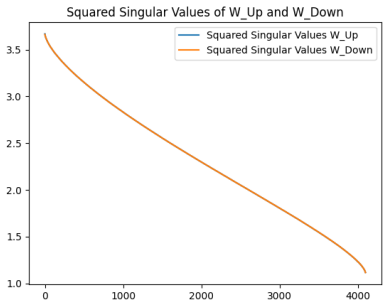


Fig. A.18.: Attention Maps *"Tarantula"* $(k = 5)$, Generated Clue: *"Large, hairy, venomous spider."*

A.5 Statistics of Singular Values in Randomized and Trained Matrices



(a) $W_q, W_k, W_v \in \mathbb{R}^{1024\times4096}$



(b) $W_{Up} \in \mathbb{R}^{14096\times4096}, W_{Down} \in \mathbb{R}^{4096\times14096}$

Fig. A.19.: Mistral 7B Squared Singular Values for initially randomized weight matrices (all layers equally initialized)

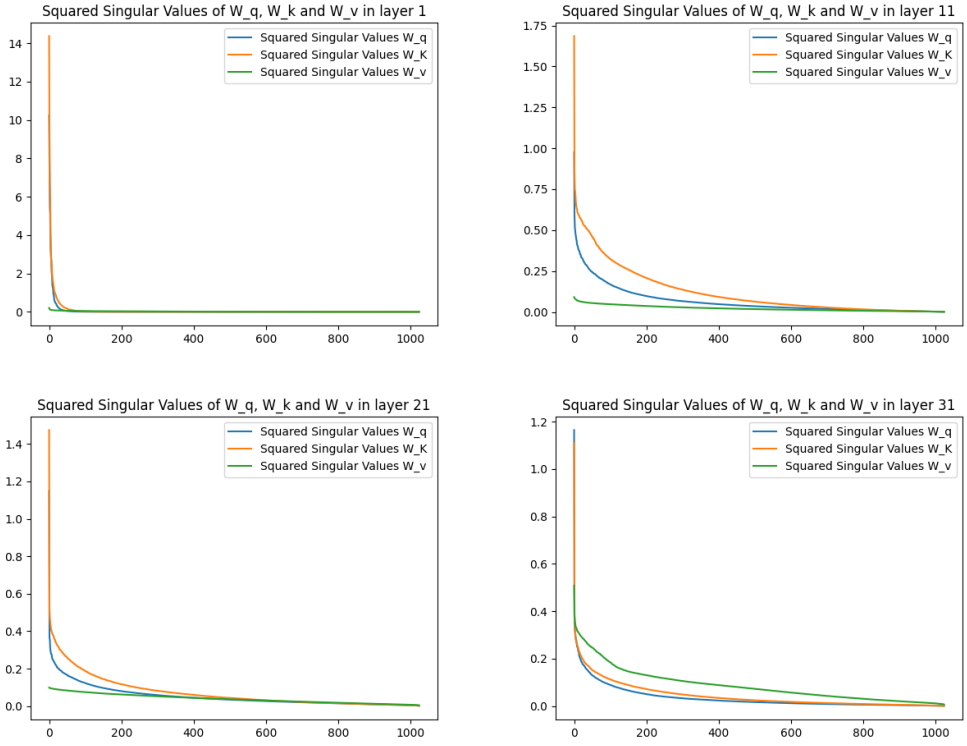


Fig. A.20.: Mistral 7B Squared Singular Values for learned $W_q, W_k, W_v \in \mathbb{R}^{1024\times4096}$

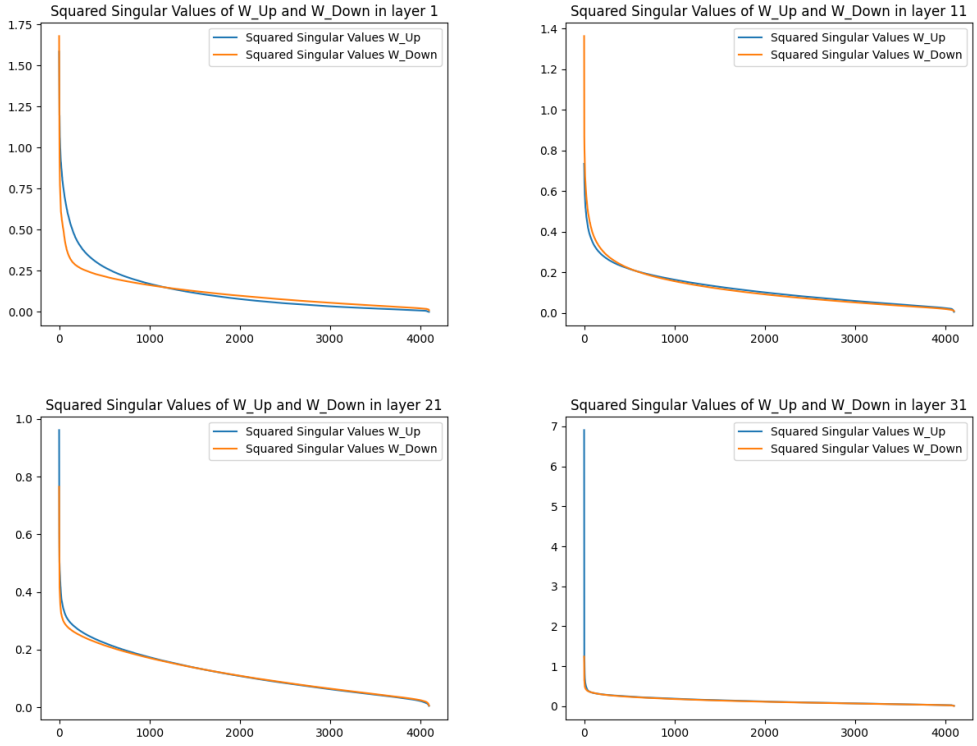


Fig. A.21.: Mistral 7B Squared Singular Values for learned $W_{Up} \in \mathbb{R}^{14096\times4096}$, $W_{Down} \in \mathbb{R}^{4096\times14096}$

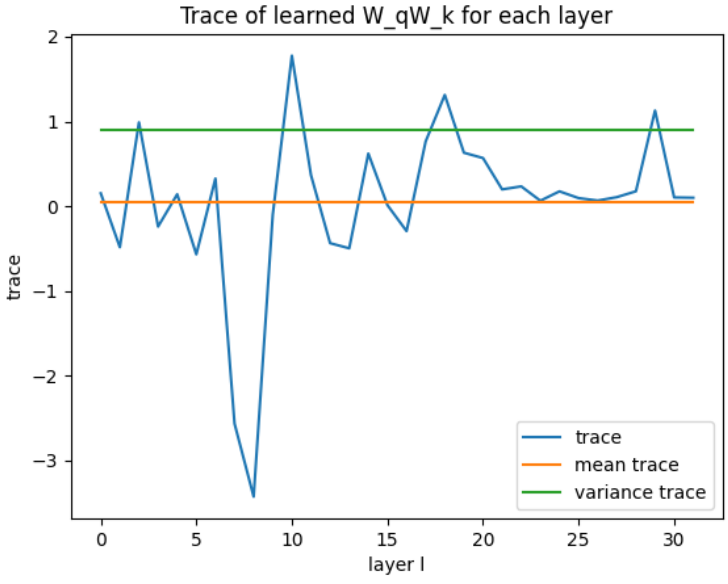


Fig. A.22.: Mistral 7B Base Alignment $\mathrm{Tr}\left(W_q^T W_k\right), \quad W_q^T W_k \in \mathbb{R}^{4096\times4096}$

B Additional Proofs

B.1 Proof Variance of Bilinear Form

Let $\begin{pmatrix} x \in \mathbb{R}^n \\ y \in \mathbb{R}^n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}\right)$ and $A, B \in \mathbb{R}^{n \times n}$, with $C = A^T B$

Then it holds:

$$\mathbb{E}[\langle Ax, By \rangle] = \mu_X^T C \mu_Y + \operatorname{Tr}(C\Sigma_{XY}) \tag{B.1}$$

And for $\Sigma_{XY} = \Sigma_{YX} = 0$:

$$\operatorname{Var}(\langle Ax, By \rangle) = \operatorname{Tr}\left(\Sigma_X C \Sigma_Y C^T\right) + \mu_Y^T C^T \Sigma_X C \mu_Y + \mu_X^T C \Sigma_Y C^T \mu_X \tag{B.2}$$

For $\Sigma_{XY} = \Sigma_{YX} = 0$:

$$\mathbb{E}[\langle Ax, By \rangle] = \mu_X^T C \mu_Y \tag{B.3}$$

Then:

$$\operatorname{Var}(\langle Ax, By \rangle) = \operatorname{Var}\left(x^T C y\right) = \mathbb{E}[\left(x^T C y\right)^2] - \mathbb{E}[x^T C y]^2 = \mathbb{E}[\left(x^T C y\right)^2] - (\mu_X^T C \mu_Y)^2 \tag{B.4}$$

We now need to show the following to prove equation B.2:

Statement:

$$\mathbb{E}[\left(x^T C y\right)^2] = \operatorname{Tr}\left(\Sigma_X C \Sigma_Y C^T\right) + \mu_Y^T C^T \Sigma_X C \mu_Y + \mu_X^T C \Sigma_Y C^T \mu_X + (\mu_X^T C \mu_Y)^2 \tag{B.5}$$

Proof: Using the cyclic property of the trace:

$$\mathbb{E}[\left(x^T C y\right)^2] = \mathbb{E}[\operatorname{Tr}\left(x^T C y y^T C^T x\right)] = \mathbb{E}[\operatorname{Tr}\left(C y y^T C^T x x^T\right)] \tag{B.6}$$

Due to the general property from "Isserlis' (Wick's) theorem" (GPT4):

$$\mathbb{E}[\operatorname{Tr}\left(C y y^T C^T x x^T\right)] = \operatorname{Tr}\left(C \mathbb{E}[y y^T] C^T \mathbb{E}[x x^T]\right) \tag{B.7}$$

The term becomes, using $E[xx^T] = \mu_X \mu_X^T + \Sigma_X$ and $E[yy^T] = \mu_Y \mu_Y^T + \Sigma_Y$:

$$\operatorname{Tr}\left(C \mathbb{E}[y y^T] C^T \mathbb{E}[x x^T]\right) = \operatorname{Tr}\left(C\left(\mu_Y \mu_Y^T + \Sigma_Y\right) C^T \left(\mu_X \mu_X^T + \Sigma_X\right)\right) \tag{B.8}$$

Splitting up into four individual terms:

$$= \mathrm{Tr}\left(C\Sigma_Y C^T\Sigma_X\right)+\mathrm{Tr}\left(C\mu_Y\mu_Y^T C^T\Sigma_X\right)+\mathrm{Tr}\left(C\Sigma_Y C^T\mu_X\mu_X^T\right)+\mathrm{Tr}\left(C\mu_Y\mu_Y^T C^T\mu_X\mu_X^T\right) \tag{B.9}$$

Cyclic property of trace results in:

$$= \mathrm{Tr}\left(\Sigma_X C\Sigma_Y C^T\right)+\mathrm{Tr}\left(\mu_Y^T C^T\Sigma_X C\mu_Y\right)+\mathrm{Tr}\left(\mu_X^T C\Sigma_Y C^T\mu_X\right)+\mathrm{Tr}\left(\mu_X^T C\mu_Y\mu_Y^T C^T\mu_X\right) \tag{B.10}$$

As the last three traces are scalar values, we finally get

$$= \mathrm{Tr}\left(\Sigma_X C\Sigma_Y C^T\right) + \mu_Y^T C^T\Sigma_X C\mu_Y + \mu_X^T C\Sigma_Y C^T\mu_X + \left(\mu_X^T C\mu_Y\right)^2 \tag{B.11}$$

$$\longrightarrow \mathrm{Var}\left(x^T Cy\right) = \mathrm{Tr}\left(\Sigma_X C\Sigma_Y C^T\right)+\mu_Y^T C^T\Sigma_X C\mu_Y+\mu_X^T C\Sigma_Y C^T\mu_X+(\mu_X^T C\mu_Y)^2-\left(\mu_X^T C\mu_Y\right)^2 \tag{B.12}$$

$$\longrightarrow \mathrm{Var}\left(x^T Cy\right) = \mathrm{Tr}\left(\Sigma_X C\Sigma_Y C^T\right) + \mu_Y^T C^T\Sigma_X C\mu_Y + \mu_X^T C\Sigma_Y C^T\mu_X \tag{B.13}$$

That concludes the proof.

C Appendix Code Sources & Usage of AI Tools

C.1 Code & ChatGPT Conversations Repositories

Code for the Experiments: "https://github.com/mwertich/BachelorThesis."

Chat-GPT4(o) Dialogues :"https://github.com/mwertich/BachelorThesisConversations."

C.2 Usage of AI Tools

AI Tool	Usage
ChatGPT4/ChatGPT4o	code generation and support for bug fixing, generation of LaTeX code
ChatGPT4/ChatGPT4o	literature research (unavailing), brainstorming, understanding advanced math theory
ChatGPT4/ChatGPT4o	math formula generation and evaluation (mentioned in the respective sections)
ChatGPT4/ChatGPT4o	grammar and spelling check, reading flow and text conciseness improvement
Grammarly	integrated into Overleaf: rough grammar and spelling check
GitHub Copilot	code generation and bug fixing in Visual Studio Code, especially for math formulas and visualization

Tab. C.1.: Overview of utilized AI tools

Declaration

I hereby declare that I have written the present thesis independently and without use of other than the indicated means. I also declare that to the best of my knowledge all passages taken from published and unpublished sources have been referenced. The thesis has not been submitted for evaluation to any other examining authority, nor has it been published in any form whatsoever. I duly noted the Regulations for Good Scientific Practice and Dealing with Scientific Misconduct.

Mainz, July 01, 2024

M. Wertich

Martin Volker Wertich