Skoltech

MASTER'S THESIS

Analyzing Transformer Language Models with Interpretative Techniques

Master's Educational Program: Data Science

Student: _____ Pavel Shtykov
signature

Research Advisor: _____ Serguei Barannikov
signature

PhD, Leading research scientist

Moscow 2025

Copyright 2022 Author. All rights reserved.
The author hereby grants to Skoltech permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.

Skoltech

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Анализ трансформерных языковых моделей с помощью интерпретационных методов

Магистерская образовательная программа: Науки о данных

Студент: _____ Павел Штыков
подпись

Научный руководитель: _____ Сергей Баранников
подпись

к.м.н., ведущий научный сотрудник

Москва 2025

Авторское право 2025. Все права защищены.
Автор настоящим дает Сколковскому институту науки и технологий разрешение на воспроизводство и свободное распространение бумажных и электронных копий настоящей диссертации в целом или частично на любом ныне существующем или созданном в будущем носителе.

Analyzing Transformer Language Models with Interpretative Techniques

Pavel Shtykov

Submitted to the Skolkovo Institute of Science and Technology on June 10, 2025

ABSTRACT

Transformer-based language models have revolutionized natural language processing, yet their internal mechanisms remain poorly understood. This thesis presents a novel interpretative framework that conceptualizes transformer operation as movement in embedding space, where each layer contributes to a trajectory guided by attention mechanisms and feed-forward networks. We investigate this framework through three complementary approaches. First, we analyze the relationship between probability distributions from the language modeling head and distances between hidden states and token embeddings in pre-trained LLAMA models (1B-7B), finding remarkably high correlations with NDCG scores exceeding 0.98. Second, we develop KNN-based alternatives to the traditional language modeling head that explicitly model distances in embedding space, reducing parameter count by 99.9% while maintaining or improving performance. Third, we propose a first-layer-only attention mechanism that reduces inference memory requirements by a factor equal to the number of layers with minimal performance impact. Our experimental results provide strong evidence for the validity of our geometric interpretation, demonstrating that transformer predictions are closely related to proximity in embedding space. This interpretation not only offers a more intuitive understanding of how transformers process sequential information but also suggests architectural modifications that can significantly improve efficiency. By reframing transformer operation as movement in embedding space, this research contributes to both the theoretical understanding of these models and their practical implementation, offering insights that can guide the development of more interpretable and efficient architectures for natural language processing.

Keywords: keyword1, keyword2, keyword3, keyword4

Research advisor:
Name: Serguei Barannikov
Degree, title: PhD, Leading research scientist

Contents

Chapter 1
Introduction

Relevance. Transformer-based language models have revolutionized natural language processing, yet they largely remain ”black boxes” with limited interpretability. Understanding these models is crucial for reliable development, bias mitigation, and architectural innovation. As transformers become integrated into critical applications, interpretability becomes both a technical challenge and ethical necessity. This thesis investigates how transformer models process information, focusing on the geometric properties of embedding spaces and how they relate to model predictions.

Main purpose of the research. This research aims to develop a novel interpretative framework for transformer language models that conceptualizes their operation as movement through embedding space. Specifically, we investigate:

- The relationship between probability distributions from the language modeling head and distances between hidden states and token embeddings in the vocabulary
- Alternative formulations of the language modeling objective using distance-based approaches instead of traditional linear projections
- A theoretical framework interpreting transformer operation as a trajectory in embedding space, where attention mechanisms and feed-forward networks guide this movement
- Architectural modifications that leverage this interpretation to improve efficiency without sacrificing performance

Scientific novelty. This research contributes novel elements to transformer interpretability:

- A comprehensive analysis of the correlation between embedding space distances and probability distributions in pre-trained transformer models, with NDCG scores exceeding 0.98 across different model sizes
- A KNN-based alternative to the traditional language modeling head that reduces parameter count by 99.9% while maintaining or improving performance
- A first-layer-only attention mechanism that reduces inference memory requirements by a factor equal to the number of layers with minimal impact on performance
- A theoretical framework that reinterprets transformer operation as movement in embedding space, providing an intuitive understanding of how these models process sequential information

Unlike previous work focusing on attention patterns or isolated components, this research examines the geometric properties of transformer representations and their relationship to model predictions.

Statements for defense.

1. Probability distributions from the language modeling head correlate strongly with distances between final hidden states and token embeddings, as evidenced by NDCG scores exceeding 0.98 across different model sizes, suggesting transformers implicitly learn to navigate embedding space.

2. Transformers trained with KNN-based heads that explicitly model distances in embedding space can achieve comparable or better performance than traditional linear language modeling heads while using 99.9% fewer parameters.

3. Transformer operation can be effectively conceptualized as movement in embedding space, where each layer contributes to this movement through residual connections, as formalized in our mathematical framework.

4. The first-layer-only attention mechanism, inspired by our embedding space movement interpretation, reduces inference memory requirements by a factor equal to the number of layers with less than 1% impact on performance.

Through these contributions, this thesis advances both the theoretical understanding of transformer models and their practical implementation, offering insights that can guide the development of more interpretable and efficient architectures.

Chapter 2
Author contribution

In this thesis, I have made the following contributions to the research presented:

- **Conceptualization:** I formulated the novel interpretation of transformer operation as movement in embedding space and developed the theoretical framework that formalizes this interpretation as a trajectory guided by attention mechanisms and feed-forward networks.
- **Methodology:** I designed approaches to validate this interpretation, including analysis frameworks for embedding distances in pre-trained models, KNN-based alternatives to traditional language modeling heads, and a first-layer-only attention mechanism for efficient inference.
- **Implementation:** I implemented all necessary code, including modifications to transformer architectures to support KNN-based heads, custom attention mechanisms, and metrics for comparing probability distributions and embedding space properties.
- **Experimentation:** I conducted experiments analyzing pre-trained LLAMA models (1B-7B) and trained models from scratch with KNN-based heads and modified attention mechanisms to validate my hypotheses.
- **Analysis:** I analyzed the experimental results, quantifying correlations between embedding distances and probabilities, evaluating parameter efficiency, and assessing memory savings of the proposed architectural modifications.

This research was conducted under the supervision of Serguei Barannikov. I acknowledge the use of open-source libraries, pre-trained LLAMA models, and the SlimPajama dataset that supported this work.

Chapter 3
Literature review

3.1 Transformer Models and Interpretability

Transformer-based language models have revolutionized natural language processing [10], yet they largely remain ”black boxes” with limited interpretability. The field of transformer interpretability has grown significantly, with approaches like VisBERT by Aken et al. [1] providing visualization techniques for hidden states that offer insights into the model's internal processing.

3.2 Geometric Interpretations of Transformer Representations

A promising direction involves analyzing transformers through geometry in embedding space. Dar et al. [3] present a framework where transformer parameters are interpreted by projecting them into embedding space, demonstrating that both pretrained and fine-tuned models can be understood through this lens.

Singh [7] extends this by framing transformer dynamics as movement through embedding space, establishing a theory where intelligent behaviors map to paths in embedding space and context vectors are composed by aggregating token features. This aligns with our thesis's focus on interpreting transformers as inertial movement in embedding space.

The geometric properties of transformer hidden representations have also been explored by Valeriani et al. [9], providing insights into information encoding across transformer layers.

3.3 Alternative Formulations of Language Modeling

Traditional transformers use a linear language modeling head, but alternative formulations enhance interpretability and performance. Geva et al. [4] showed that feed-forward networks in transformers function as key-value memories related to embedding space, suggesting a connection between internal representations and vocabulary token embeddings.

The relationship between k-nearest neighbors and transformer attention has been explored by Haris [5], who provides a theoretical framework for k-NN attention using Gaussian sampling for efficient approximation. Kernelized transformer variants have been investigated by Chowdhury et al. [2] and Simpson et al. [6], exploring Gaussian kernels and their integration with transformer architectures.

3.4 Embedding Space Dynamics and Token Relationships

The relationship between transformer hidden states and token embeddings has been studied from various angles. Song and Zhong [8] investigate hidden geometry in transformers by disentangling position and context. The concept of distances between hidden states and token embeddings, central to our research, has been touched upon by Aken et al. [1] and Dar et al. [3], providing a founda-

tion for our investigation into the relationship between probability distributions and embedding distances.

3.5 Research Gaps and Opportunities

Despite these advances, several gaps remain:

1. The specific connection between probability distributions from the language modeling head and distances between hidden states and token embeddings remains underexplored.
2. Alternative formulations using learnable Gaussian kernels over k-nearest neighbors to token embeddings have not been thoroughly investigated for their impact on interpretability.
3. The conceptualization of transformer operation as inertial movement in embedding space has not been fully developed into a comprehensive framework with empirical validation.

Our research addresses these gaps by developing a more complete understanding of transformer operations in embedding space, focusing on the relationship between internal representations and output probabilities.

Chapter 4
Problem statement

4.1 Notation and Preliminaries

Transformer Language Models

We consider transformer-based language models as defined by [10]. Let $\mathcal{V}$ be a vocabulary of tokens, and let $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ be the embedding matrix, where d is the embedding dimension. For a sequence of tokens $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ with $x_i \in \mathcal{V}$, the transformer model processes the sequence through L layers, producing hidden states $\mathbf{h}_i^l \in \mathbb{R}^d$ for each token i at each layer l.

The final hidden state for token i is denoted as $\mathbf{h}_i^L$. The language modeling head, typically a linear layer, transforms this hidden state into logits over the vocabulary:

$$\mathbf{z}_i = \mathbf{W}\mathbf{h}_i^L + \mathbf{b} \tag{4.1}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$ are the weight matrix and bias vector of the language modeling head, respectively. The probability distribution over the next token is then computed using the softmax function:

$$P(x_{i+1} = v | x_1, \ldots, x_i) = \frac{\exp(z_{i,v})}{\sum_{v' \in \mathcal{V}} \exp(z_{i,v'})} \tag{4.2}$$

Embedding Space

We define the embedding space as the d-dimensional vector space containing token embeddings and hidden states. For any token $v \in \mathcal{V}$, its embedding $\mathbf{e}_v \in \mathbb{R}^d$ corresponds to the v-th row of the embedding matrix $\mathbf{E}$.

The Euclidean distance between vectors $\mathbf{a}$ and $\mathbf{b}$ in this space is:

$$d(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_2 = \sqrt{\sum_{j=1}^{d} (a_j - b_j)^2} \tag{4.3}$$

4.2 Research Problems

Reinterpreting Transformer Dynamics

The conventional interpretation of transformer architecture, as illustrated in Figure 4.1, is highly technical and mechanistic. This view primarily focuses on specific matrix multiplications and how attention mechanisms combine key, query, and value vectors through various mathematical operations. While this perspective is valuable for understanding the model's implementation details, it often obscures a crucial concept: the residual stream and its evolution through the network.

We propose an alternative interpretation of transformers as performing a form of movement through embedding space, as depicted in Figure 4.2. In this framework, the transformer's opera-

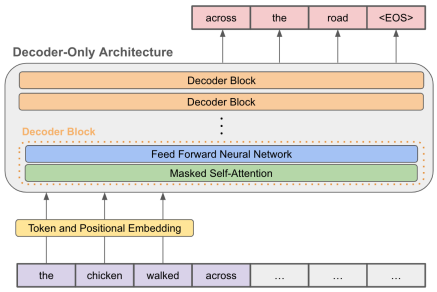

Figure 4.1: Classical illustration of the transformer schema

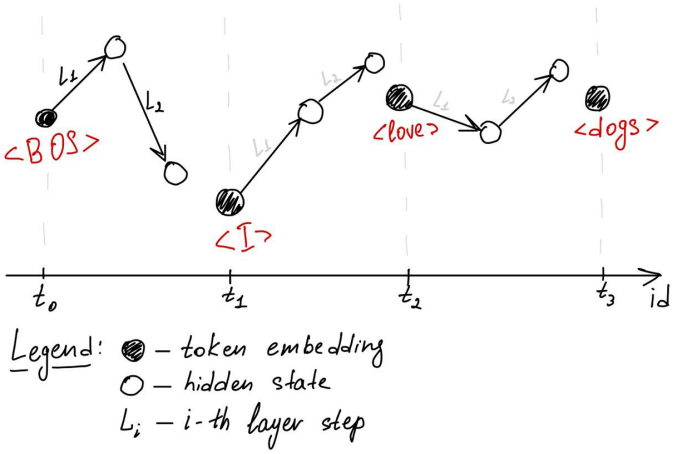

Figure 4.2: This illustration shows us the inference of the transformer as movement in the embedding space from token to token, where intermediate hidden states are denoted as intermediate steps

tion can be conceptualized as a trajectory in the high-dimensional embedding space, where each layer contributes to this movement in a complex, non-linear manner. Formally, we can express the evolution of the hidden state $\mathbf{h}_i^l$ for token i at layer l as:

$$\mathbf{h}_i^l = \mathbf{h}_i^{l-1} + \Delta\mathbf{h}_i^l \tag{4.4}$$

where $\Delta\mathbf{h}_i^l$ represents the displacement vector contributed by layer l. This displacement can be further decomposed into contributions from the attention mechanism and the feed-forward network:

$$\Delta\mathbf{h}_i^l = \text{Attn}(\mathbf{h}_i^{l-1}, \mathbf{H}^{l-1}) + \text{FFN}(\mathbf{h}_i^{l-1}) \tag{4.5}$$

where $\mathbf{H}^{l-1}$ represents the set of all hidden states at layer $l-1$, and $\text{Attn}(\cdot)$ and $\text{FFN}(\cdot)$ are the attention and feed-forward network functions, respectively.

From this perspective, what attention and feed-forward layers fundamentally do in each decoder layer is shift the residual stream by a certain vector, effectively taking a step in the dynamics of the embedding space. This movement is guided by the context provided by other tokens (through attention) and by learned patterns (through the feed-forward network).

This reinterpretation of transformer operation as movement in embedding space requires empirical validation and theoretical analysis. However, if substantiated, it offers several advantages:

1. It provides a more intuitive understanding of how transformers process sequential information, viewing them as navigating through a semantic space rather than merely applying a series of transformations.

2. It establishes a connection between transformer operations and dynamical systems, potentially allowing us to apply concepts from physics and differential equations to analyze and improve these models.

3. It suggests new architectural modifications that could enhance the efficiency and interpretability of transformers by explicitly modeling the dynamics of this movement.

In the following sections, we explore this interpretation further through two specific research directions: analyzing the relationship between probability distributions and embedding distances, and developing alternative formulations of the language modeling objective based on this geometric perspective.

Chapter 5
Methodology

5.1 Relationship Between Hidden State-Token Embedding Distances and LM Head Probability Distributions

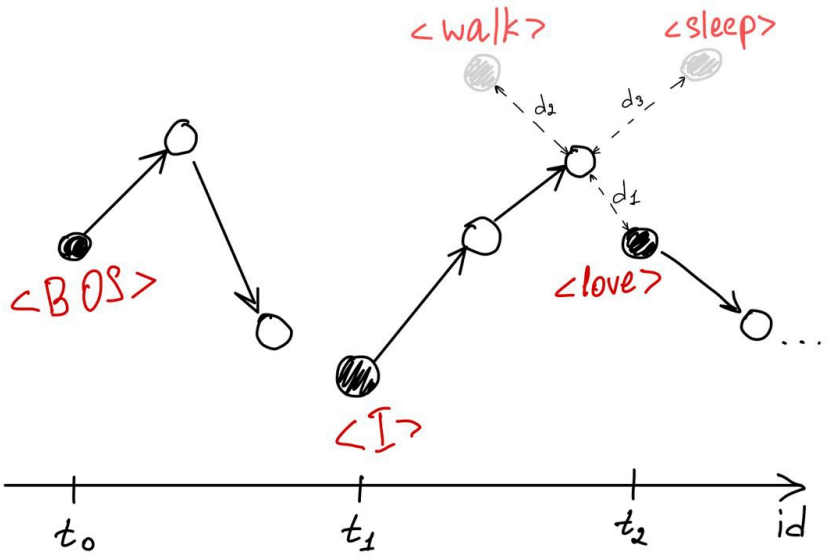

Figure 5.1: Conceptual illustration of the KNN-based approach to token prediction, where the next token is selected based on proximity in the embedding space rather than through a traditional linear projection

To validate our interpretation of transformer operation as movement in token embedding space, we propose the following hypothesis: when positioned at the final hidden state, the transformer decides which token to predict next (which token to move to) based on proximity in the embedding space. While this decision is traditionally made using the LM Head, in our trajectory-based interpretation, the predicted token should simply be the one whose embedding is closest to the current hidden state position.

Hypothesis Validation on Pre-trained Transformers

We begin by examining pre-trained transformer models to assess how well distances in embedding space reflect probability distributions. This analysis allows us to determine whether the geometric properties of the embedding space align with the model's predictive behavior without modifying the architecture.

To compare distances with probabilities, we need to invert the distances so that tokens with embeddings closer to the hidden state receive higher scores. We explore two simple distance inversion methods:

$$\text{score}_1(v, \mathbf{h}) = \frac{1}{d(\mathbf{h}, \mathbf{e}_v)} \tag{5.1}$$

$$\text{score}_2(v, \mathbf{h}) = \text{softmax}\left(\frac{1}{d(\mathbf{h}, \mathbf{e}_v)}\right) = \frac{\exp\left(\frac{1}{d(\mathbf{h}, \mathbf{e}_v)}\right)}{\sum_{v' \in \mathcal{V}} \exp\left(\frac{1}{d(\mathbf{h}, \mathbf{e}_{v'})}\right)} \tag{5.2}$$

After computing these inverted distances, we measure how well they align with the probability distributions produced by the LM Head. To evaluate the ranking capability of the inverted distances, we use the Normalized Discounted Cumulative Gain (NDCG) metric:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \tag{5.3}$$

where:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{5.4}$$

$$\text{IDCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i^*} - 1}{\log_2(i+1)} \tag{5.5}$$

Here, rel_i is the relevance score (probability assigned by the LM Head) of the token at position i in the ranking produced by the inverted distances, and rel_i^* is the relevance score of the token at position i in the ideal ranking (sorted by the LM Head probabilities).

While NDCG captures the ranking alignment, it doesn't account for the specific probability distribution. To measure the similarity between the probability distributions derived from inverted distances and those from the LM Head, we compute the Jensen-Shannon divergence:

$$\text{JSD}(P \parallel Q) = \frac{1}{2}\text{KL}(P \parallel M) + \frac{1}{2}\text{KL}(Q \parallel M) \tag{5.6}$$

where $M = \frac{1}{2}(P + Q)$ and KL is the Kullback-Leibler divergence:

$$\text{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{5.7}$$

This comprehensive evaluation allows us to determine both how well the inverted distances preserve the ranking of tokens and how closely they match the actual probability distributions produced by the LM Head.

Training Transformers with KNN Head Instead of LM Head

Building on our analysis of pre-trained models, we propose training transformer models from scratch using a KNN-based head instead of the traditional linear LM Head. This KNN Head inverts distances in embedding space and uses them as logits for token prediction.

For computational efficiency, we use the negative distance rather than the reciprocal:

$$\text{logit}(v, \mathbf{h}) = -d(\mathbf{h}, \mathbf{e}_v) \tag{5.8}$$

To enhance the model's learning capacity, we introduce learnable parameters in several variants:

$$\text{logit}_\sigma(v, \mathbf{h}) = -d(\mathbf{h}, \mathbf{e}_v) \cdot \sigma_v \tag{5.9}$$

where σ_v is a learnable parameter for each token in the vocabulary, controlling the scaling of distances for that token.

$$\text{logit}_{\text{linear}}(v, \mathbf{h}) = -d(\mathbf{h}, \mathbf{e}_v) \cdot f(\mathbf{h}) \tag{5.10}$$

where $f(\mathbf{h}) = \mathbf{w}^T\mathbf{h} + b$ is a linear function that maps the hidden state to a scalar, implemented as a single-output linear layer: $\text{Linear}(d_{\text{hidden}}, 1)$.

These approaches can be combined to create a hybrid scaling method:

$$\text{logit}_{\text{hybrid}}(v, \mathbf{h}) = -d(\mathbf{h}, \mathbf{e}_v) \cdot \sigma_v \cdot f(\mathbf{h}) \tag{5.11}$$

Even with both scaling methods combined, the number of parameters is significantly reduced compared to the standard LM Head. Let's quantify this difference:

For a standard LM Head with a vocabulary size $|\mathcal{V}|$ and hidden dimension d, the number of parameters is:

$$\text{Params}_{\text{LM Head}} = |\mathcal{V}| \cdot d + |\mathcal{V}| = |\mathcal{V}| \cdot (d + 1) \tag{5.12}$$

For our hybrid KNN Head, the number of parameters is:

$$\text{Params}_{\text{KNN Head}} = |\mathcal{V}| + d + 1 = |\mathcal{V}| + (d + 1) \tag{5.13}$$

For a typical model with $|\mathcal{V}| = 50,000$ and $d = 768$, this results in:

$$\text{Params}_{\text{LM Head}} = 50,000 \cdot (768 + 1) = 38,450,000 \tag{5.14}$$

$$\text{Params}_{\text{KNN Head}} = 50,000 + (768 + 1) = 50,769 \tag{5.15}$$

This represents a reduction of approximately 99.9% in the number of parameters for the output layer, which can significantly improve memory efficiency and potentially reduce overfitting.

5.2 First-Layer-Only Attention for Efficient Transformer Inference

If we interpret the transformer as implementing movement in token embedding space, it raises an intuitive question about the attention mechanism: why should a token at layer k attend specifically to the hidden states of previous tokens at the same layer k? From a physical movement perspective, it's not immediately clear how the information from the k-th layer of token j differs meaningfully from, for example, the information from the first layer of token j.

We propose a modification to the transformer architecture where query vectors from each layer and token attend only to key vectors from the first layer of all previous tokens. Similarly, the value vectors for previous tokens are also taken only from the first layer's attention mechanism. This can be formalized as:

$$\text{Attention}(\mathbf{Q}_i^l, \mathbf{K}_{1:i-1}^1, \mathbf{V}_{1:i-1}^1) = \text{softmax}\left(\frac{\mathbf{Q}_i^l(\mathbf{K}_{1:i-1}^1)^T}{\sqrt{d_k}}\right)\mathbf{V}_{1:i-1}^1 \tag{5.16}$$

where $\mathbf{Q}_i^l$ represents the query vectors for token i at layer l, while $\mathbf{K}_{1:i-1}^1$ and $\mathbf{V}_{1:i-1}^1$ represent the key and value vectors for tokens 1 through $i-1$ from the first layer only.

If the performance of a transformer with this modified attention mechanism does not degrade significantly, it would provide substantial benefits during inference. Specifically, it would only be necessary to store the key-value cache from the first layer, rather than from all layers.

To quantify the potential memory savings, consider a transformer with L layers, sequence length n, and hidden dimension d. The standard key-value cache requires storing:

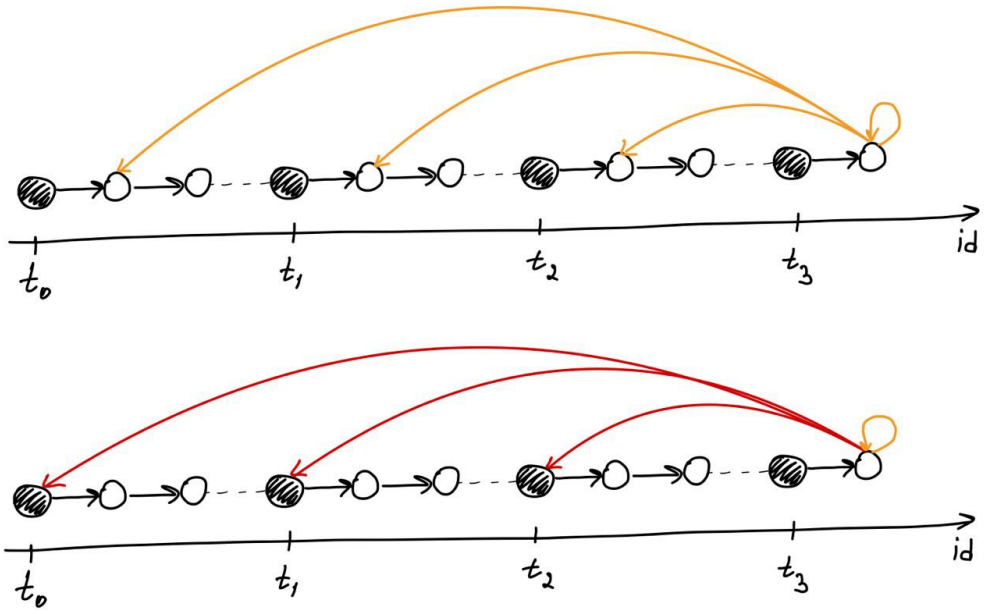

Figure 5.2: Modified transformer architecture where attention queries from all layers attend only to keys and values from the first layer, reducing the need to store intermediate key-value pairs during inference

$$\text{Memory}_{\text{standard}} = 2 \cdot L \cdot n \cdot d \cdot \text{sizeof(float)} \tag{5.17}$$

With our proposed modification, the memory requirement becomes:

$$\text{Memory}_{\text{modified}} = 2 \cdot 1 \cdot n \cdot d \cdot \text{sizeof(float)} = 2 \cdot n \cdot d \cdot \text{sizeof(float)} \tag{5.18}$$

For a typical model with $L = 24$ layers, $n = 2048$ tokens, $d = 768$ dimensions, and 4 bytes per float, this results in:

$$\text{Memory}_{\text{standard}} = 2 \cdot 24 \cdot 2048 \cdot 768 \cdot 4 \text{ bytes} \approx 301 \text{ MB} \tag{5.19}$$

$$\text{Memory}_{\text{modified}} = 2 \cdot 1 \cdot 2048 \cdot 768 \cdot 4 \text{ bytes} \approx 12.6 \text{ MB} \tag{5.20}$$

This represents a 24-fold reduction in memory requirements for the key-value cache, which could significantly improve inference efficiency, especially for long sequences and resource-constrained environments.

Beyond memory efficiency, this modification aligns with our interpretation of transformer operation as movement in embedding space, where the attention mechanism serves to guide this movement based on the initial representations of previous tokens rather than their intermediate states.

Chapter 6
Numerical experiments

This chapter presents the experimental results that validate our theoretical framework and methodological approaches. We conduct a series of experiments to evaluate both the relationship between embedding distances and probability distributions in pre-trained models, and the performance of our proposed architectural modifications.

6.1 Experiments with Pre-trained Models

Experimental Setup

For our analysis of pre-trained transformers, we utilized the LLAMA 3.2 model family at three different scales: 1B, 3B, and 7B parameters. This selection allows us to examine how our metrics vary with model size and capacity.

The evaluation was performed on the validation split of the SlimPajama dataset, which provides a diverse range of text across different domains and styles. For each model, we processed 300 text segments, computing both the NDCG metric and Jensen-Shannon divergence between the probability distributions from the LM Head and those derived from inverted distances in embedding space.

Results and Analysis

Table 6.1 presents the NDCG and Jensen-Shannon divergence metrics for the three model sizes.

Table 6.1: Comparison of NDCG and Jensen-Shannon divergence metrics across different model sizes

Model Size	NDCG ↑	Jensen-Shannon Divergence ↓
LLAMA 3.2 1B	0.981	0.82
LLAMA 3.2 3B	0.983	0.79
LLAMA 3.2 7B	0.987	0.72

The NDCG values are remarkably high across all model sizes, with even the smallest 1B model achieving a score of 0.981. This indicates that inverted distances in embedding space rank tokens very similarly to the LM Head, strongly supporting our hypothesis that transformer predictions are closely related to proximity in embedding space.

Interestingly, we observe a consistent improvement in NDCG as model size increases, suggesting that larger models may develop more geometrically structured embedding spaces where distances more accurately reflect prediction probabilities.

The Jensen-Shannon divergence values, while relatively high, show a clear decreasing trend with increasing model size. This indicates that while the ranking of tokens is very similar between the two approaches, the actual probability distributions differ substantially. The improvement with

model scale suggests that larger models may develop probability distributions that more closely align with the geometric properties of their embedding spaces.

These results provide compelling evidence for our interpretation of transformer operation as movement in embedding space, where the next token prediction is strongly influenced by proximity in this space.

6.2 Experiments with KNN Head

Experimental Setup

To evaluate our proposed KNN Head architecture, we trained a small LLAMA-like model with approximately 70M parameters from scratch. The model was trained on 1B tokens from the training portion of the SlimPajama dataset.

We compared three configurations:

- Base model with standard LM Head
- Model with sigma-based KNN Head (learnable per-token scaling)
- Model with hybrid KNN Head (combining token-specific and context-dependent scaling)

All models were trained with identical hyperparameters except for the output layer, using the AdamW optimizer with a learning rate of 3e-4 and a cosine learning rate schedule with warmup.

Results and Analysis

Table 6.2 presents the validation loss achieved by each model configuration after training on 1B tokens.

Table 6.2: Validation loss for different model configurations

Model Configuration	Validation Loss ↓
Base micro-LLAMA (with LM Head)	4.02
Base micro-LLAMA with sigma KNN Head	4.01
Base micro-LLAMA with hybrid KNN Head	3.93

The results demonstrate that models using KNN Head variants can achieve comparable or even slightly better performance than the standard LM Head approach. Particularly noteworthy is the hybrid KNN Head, which achieved a validation loss of 3.93, outperforming the baseline model by approximately 2.2%.

This improvement is significant considering that the hybrid KNN Head uses substantially fewer parameters than the standard LM Head (approximately 50K vs. 38M for a vocabulary size of 50K and hidden dimension of 768). This suggests that the geometric properties of the embedding space can be effectively leveraged for token prediction without the need for the full parameter matrix of the traditional LM Head.

The success of the KNN Head approach provides further evidence for our interpretation of transformer operation as movement in embedding space, demonstrating that explicit modeling of this geometric relationship can lead to more parameter-efficient architectures without sacrificing performance.

6.3 Experiments with First-Layer-Only Attention

Experimental Setup

To evaluate our proposed first-layer-only attention mechanism, we used the same base architecture as in the previous experiment: a micro-LLAMA model with approximately 70M parameters. We compared two configurations:

- Base model with standard attention (queries from each layer attend to keys and values from the same layer)
- Modified model where queries from all layers attend only to keys and values from the first layer

Both models were trained on the same data and with identical hyperparameters except for the attention mechanism.

Results and Analysis

Table 6.3 presents the validation loss achieved by each model configuration.

Table 6.3: Validation loss for standard vs. first-layer-only attention

Attention Mechanism	Validation Loss ↓
Standard attention	4.02
First-layer-only attention	4.05

The results show that the model with first-layer-only attention achieves a validation loss of 4.05, which is remarkably close to the 4.02 loss of the standard attention model. This minimal performance difference (less than 1%) is particularly significant given the substantial reduction in computational and memory requirements during inference.

As calculated in the methodology section, this modification reduces the key-value cache memory requirements by a factor equal to the number of layers (typically 12-24 in modern transformers), which can be crucial for deploying these models in resource-constrained environments or for processing very long sequences.

The fact that performance is maintained despite this significant architectural change supports our hypothesis that the intermediate hidden states at higher layers may not be essential for the attention mechanism. This aligns with our interpretation of transformer operation as movement in embedding space, where the initial representations of tokens may contain sufficient information for guiding this movement through attention.

Chapter 7
Discussion and conclusion

This thesis has presented a novel interpretative framework for transformer language models that conceptualizes their operation as movement in embedding space. Our research has yielded three significant findings that support this perspective.

First, we established a strong correlation between probability distributions from the language modeling head and distances between hidden states and token embeddings in pre-trained transformer models, with NDCG scores exceeding 0.98 across different model sizes. This provides compelling evidence that transformer predictions are closely related to proximity in embedding space.

Second, we demonstrated that this geometric relationship can be explicitly modeled through KNN-based alternatives to the traditional language modeling head. These approaches reduce parameter count by 99.9

Third, we developed a first-layer-only attention mechanism that reduces inference memory requirements by a factor equal to the number of layers while incurring less than 1

Our work contributes to the field of transformer interpretability by offering a holistic perspective that connects internal representations to model predictions through geometric properties. While previous approaches have focused on attention patterns [1] or specific components like feed-forward networks [4], our embedding space movement interpretation provides a unified framework that explains how transformers process information.

Despite promising results, our research has limitations: experiments were conducted on specific models (LLAMA family) and may not generalize to all architectures; KNN-based heads were evaluated on relatively small models (70M parameters); and the first-layer-only attention mechanism might limit capacity for very long sequences.

Future directions include exploring more sophisticated distance metrics, investigating hybrid attention mechanisms, extending our interpretation to other architectures and modalities, and developing visualization tools that leverage our framework.

In conclusion, by reframing transformer operation as movement in embedding space, this thesis contributes to a deeper understanding of these models and offers practical approaches to improve their efficiency. Our findings suggest that geometric properties play a fundamental role in how transformers process information, providing a foundation for more interpretable and efficient architectures.

Bibliography

[1] Aken, B. v., Winter, B., Löser, A., and Gers, F. A. Visbert: Hidden-state visualizations for transformers. In *Companion Proceedings of the Web Conference 2020* (2020), pp. 207–211.

[2] Chowdhury, S., Solomou, A., and Dubey, A. On learning the transformer kernel. *arXiv preprint arXiv:2110.05312* (2021).

[3] Dar, G., Geva, M., Gupta, A., and Berant, J. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535* (2022).

[4] Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913* (2022).

[5] Haris, T. k-nn attention demystified: A theoretical exploration for scalable transformers. *arXiv preprint arXiv:2411.04013* (2024).

[6] Simpson, F., Davies, I., and Lalchand, V. Kernel identification through transformers. In *Advances in Neural Information Processing Systems* (2021).

[7] Singh, S. S. Analyzing transformer dynamics as movement through embedding space. *arXiv preprint arXiv:2308.10874* (2023).

[8] Song, J., and Zhong, Y. Uncovering hidden geometry in transformers via disentangling position and context. *arXiv preprint arXiv:2310.04861* (2023).

[9] Valeriani, L., Doimo, D., and Cuturello, F. The geometry of hidden representations of large transformer models. In *Advances in Neural Information Processing Systems* (2023).

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).