Computation Mechanism Behind LLM Position Generalization

Anonymous ACL submission

Abstract

Most written natural languages are composed of sequences of words and sentences. Similar to humans, large language models (LLMs) exhibit flexibility in handling textual positions - a phenomenon we term **position generalization**. They can understand texts with position perturbations and generalize to longer texts than those encountered during training with the latest techniques. These phenomena suggest that LLMs handle positions in a tolerant manner, but how LLMs computationally process positional relevance remains largely unexplored. In this work, we show how LLMs enforce certain computational mechanisms to allow for the aforementioned tolerance in position perturbations. Despite the complex design of the self-attention mechanism, this work reveals that LLMs learn a counterintuitive disentanglement of attention logits. Their values show a 0.959 linear correlation with an approximation of the arithmetic sum of positional relevance and semantic importance. Furthermore, we identify a prevalent pattern in intermediate features that enables this effect, suggesting that it is a learned behavior rather than a natural result of the model architecture. Based on these findings, we provide computational explanations and criteria for the aforementioned position flexibilities observed in LLMs.

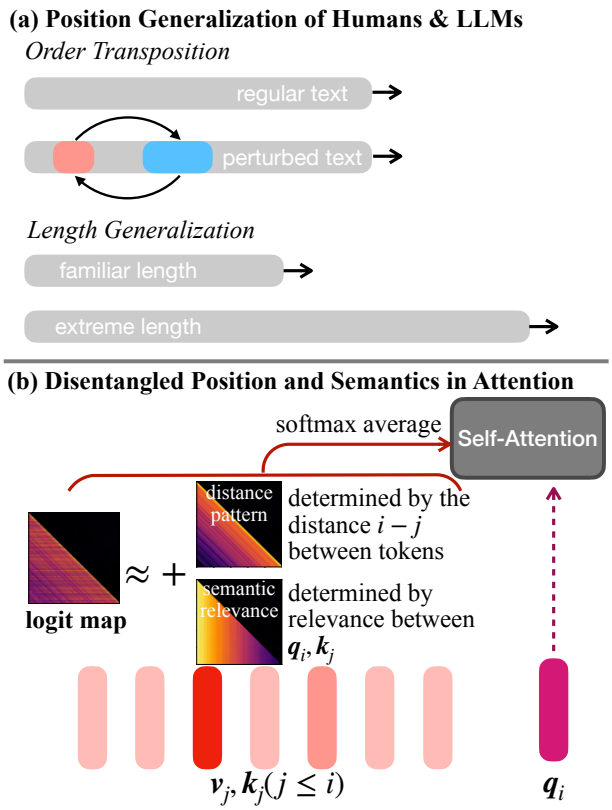


Figure 1: **(a)** LLMs, like humans, exhibit position generalization in various forms. **(b)** Self-attention in LLMs disentangles positional and semantic relevance, so as not to be sensitive over position perturbations. The "distance pattern" and "semantic relevance" shows two subcomponents of the logit map that depend on positional and semantic relation between tokens, respectively.

1 Introduction

Most natural languages are written as sequences of textual elements such as characters, words, and sentences. Despite this sequential nature, large language models (LLMs) exhibit remarkable tolerance in handling textual positions, just as observed in human studies (Bruner and O'Dowd, 1958; Rawlinson, 2007). LLMs can comprehend text with position perturbations (Sinha et al., 2021b; Pham et al., 2021) and generalize to longer sequences than those seen during training with techniques like LM-Infinite (Han et al., 2024) and InfLLM (Xiao et al., 2024). These raise the question of how positional relevance is handled internally. While prior research has explored various positional encoding strategies (Su et al., 2021; Press et al., 2021), the underlying computational mechanisms of LLMs' position robustness remain largely unexplored.

In this work, we analyze the self-attention mechanism of modern LLMs to investigate how they process positional relevance information. Our findings reveal that LLMs learn a counter-intuitive *disentanglement* in attention logits (Sec 3.1, 3.2). With a

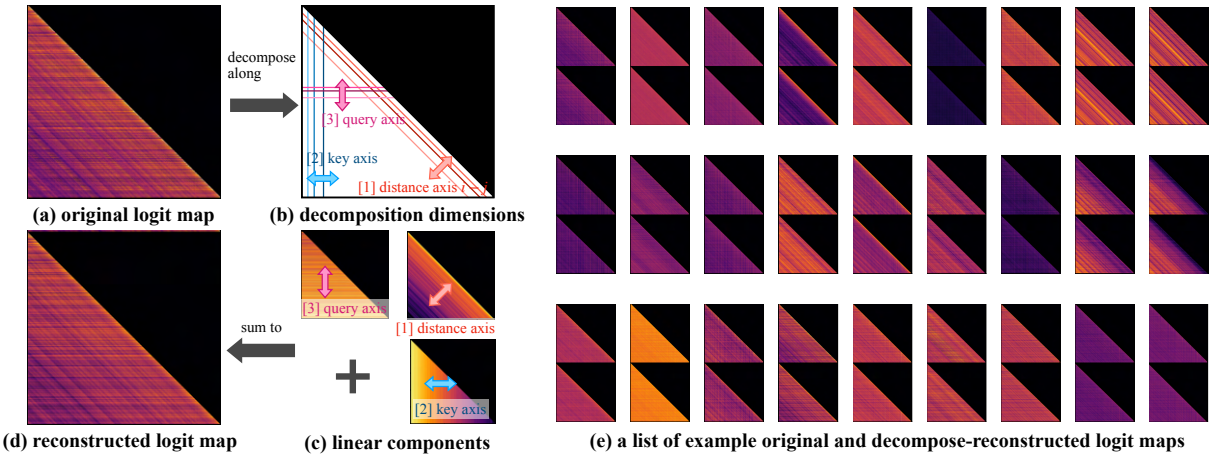


Figure 2: As a starting point of our study, we find that a 3-axis linear approximation is surprisingly similar to the original attention logit maps. Fig (e) is a set of original logit maps (upper ones) and their constructions (lower ones). More details are in Sec 3.1.

linear sum of two components $f(\boldsymbol{q}, i-j)+g(\boldsymbol{q}, \boldsymbol{k})$, which are about positional relation $i-j$ and semantical relation $g(\boldsymbol{q}, \boldsymbol{k})$, respectively, the attention logits can be approximated with >0.95 linear correlation. Furthermore, we identify a systematic pattern in intermediate representations that enables this effect (Sec 3.3), suggesting that it is a learned behavior rather than an inherent consequence of model architecture.

Finally, we apply these findings to provide a computational explanation for the position generalization phenomenon in LLMs (Sec 4). We demonstrate how text order transpositions on up to 5% of all words only marginally affect the LLM's perplexity and downstream performance. This linguistic observation can be simulated by transposing the order of hidden features or perturbing the positional indices in relative position encoding, suggesting an analogy between human behaviors and the LLM computational mechanism. We further explain how length generalization techniques can extend LLMs to extreme lengths without parameter updates. Taking insights that attention logits are approximately linearly additive terms over a pool of features, we show how self-attention is relatively tolerating while still ensuring the attention output vectors $\boldsymbol{o}$ fall within the training-time distribution. This explains how feature distribution shift is avoided in length generalization techniques.

2 Related Work and Background

2.1 Self-Attention and Positional Encoding

Self-attention is the core design in most modern LLMs for information flow to words from their contexts (Vaswani et al., 2017). It is also the primary (and often the only) component to inject text position information since the introduction of relative position encoding (Su et al., 2021; Touvron et al., 2023; Dubey et al., 2024; OpenAI, 2023), which is the subject of investigation in this work. Despite architecture variants, it is generally designed as a Softmax-based weighted average over contextual "value" vectors $\{\boldsymbol{v}_j | j \leq i\}$ before current position i. The average weights $w(\boldsymbol{q}_i, \boldsymbol{k}_j, i-j)$ are determined by the relevance between the current word's "query" vector $\boldsymbol{q}_i$, contextual words' "key" vectors $\{\boldsymbol{k}_j | j \leq i\}$, and their relative position $i-j$. The output feature vector for the current token $\boldsymbol{o}_i$ is therefore:

$$\boldsymbol{o}_i = \sum_{j \leq i} \frac{w(\boldsymbol{q}_i, \boldsymbol{k}_j, i-j)}{\sum_{j' \leq i} \exp w(\boldsymbol{q}_i, \boldsymbol{k}_{j'}, i-j')} \boldsymbol{v}_j \quad (1)$$

. In spite of the existence of other choices of function $w(\cdots)$ like Alibi (Press et al., 2021), the defacto mainstream choice is RoPE (Su et al., 2021). It decomposes $\boldsymbol{q}$ and $\boldsymbol{k}$ vectors into 2-D tuples and lets them rotate in angle $(i-j)\theta_r$, where each 2-D tuple r has a different rotating "angular speed" θ_r.

2.2 Position Generalization (of both LLMs and Humans)

Both humans and LLMs exhibit the ability to understand language with variable word or sentence positions. This phenomenon is related to multiple concepts from different perspectives. Although written languages are usually represented as sequences of textual elements (such as characters, words, and sentences), they differ in their

Word Order Flexibility (Bakker, 1998; Kaiser and Trueswell, 2004). Some languages (e.g., English, Chinese, Vietnamese, Indonesian) require a strict word order, while others (e.g., Hungarian, Japanese and Latin) allow more flexibility in order, which encodes pragmatic information such as emphasis (Payne, 1992). This aspect has been computationally measured (Kahane et al., 2023) and used to evaluate linguistic complexity (Szmrecsanyi, 2016).

Nevertheless, even when texts are perturbed to the extent that they no longer conform to regular language, humans can still understand them under certain conditions. The **Transposed Letter Effect** (Bruner and O'Dowd, 1958; Rawlinson, 2007) describes the ability to understand texts when the letter order is scrambled within words. Language models also demonstrate the ability to perform downstream tasks on syntactically scrambled inputs, as shown in **Unnatural Language Processing** (Sinha et al., 2021b; Pham et al., 2021). Sinha et al. (2021a) report comparable or improved quality of masked language models after pre-training on such corpora. At the sentence level, models pre-trained on randomly ordered corpora show improved performance on tasks involving complex contextual reasoning (Shi et al., 2024).

Preliminary studies on neural mechanisms underlying these phenomena in humans have been conducted in cognitive neuroscience (Garcia-Orza et al., 2010; Duñabeitia et al., 2012; Carreiras et al., 2015), showing prevalent while varying robustness to transposition effects on letters, digits, and symbols in human brains. This work offers a computational counterpart, interpreting how position generalization is reflected in the internal mechanism of LLMs.

3 LLMs Disentangle Position and Semantics in Attention

How do LLMs handle the interaction between positional relation and semantic relation? The attention logit function does not need to be smooth or simple across distances. It can be designed with arbitrary complexity so that at every distance $i-j$, the function $w(i-j,\cdot,\cdot)$ behaves drastically differently. In fact, RoPE adopts a complex design that could theoretically implement (inverse) discrete Fourier transform, allowing it to approximate arbitrary functions with a sufficiently large dimension size. Unless otherwise stated, we use Llama-3.2-7B model as subject of study, and extend results to other models in Appendix.

Counter-intuitively, in this section, we reveal that LLMs learn a special feature pattern to empirically simplify the logit function $w(\cdots)$. The resulting attention logits can be approximately disentangled as an arithmetic addition of position relevance (determined by $i-j$ and $\boldsymbol{q}_i$) and semantic importance (determined by $\boldsymbol{k}_j$). We will start with an interesting observation of low-rank components in attention maps in Sec 3.1. Taking it as an inspiration, Sec 3.2 shows how the attention logits can be approximately disentangled into position and semantic-related components. Finally, Sec 3.3 shows how LLMs computationally achieve this mechanism by enforcing a special pattern in key and query vectors.

3.1 Starting Point: 3-Axis Linear Approximation of Logit Matrix

Let us start by looking at an attention head's logit matrix $W \in \mathcal{R}^{n\times n}$ with elements $W_{i,j} = w(\boldsymbol{q}_i, \boldsymbol{k}_j, i-j)$ in Fig 2(a). It is lower-triangular in causal language models where only past tokens are within the attention scope of the current token. Despite combining information of three variables $i-j, \boldsymbol{q}_i, \boldsymbol{k}_j$, there are visible 1-d patterns along horizontal, vertical, and (off-)diagonal axes. These three axes are coincidentally the ones associated with the three variables as depicted in Fig 2(b): in axis 1, $i-j$ does not vary in a diagonal line (the "distance axis"); in axis 2, $\boldsymbol{k}_j$ and j do not vary in a vertical column (the "key axis"); in axis 3, $\boldsymbol{q}_i$ and i do not vary in a horizontal row (the "query axis").

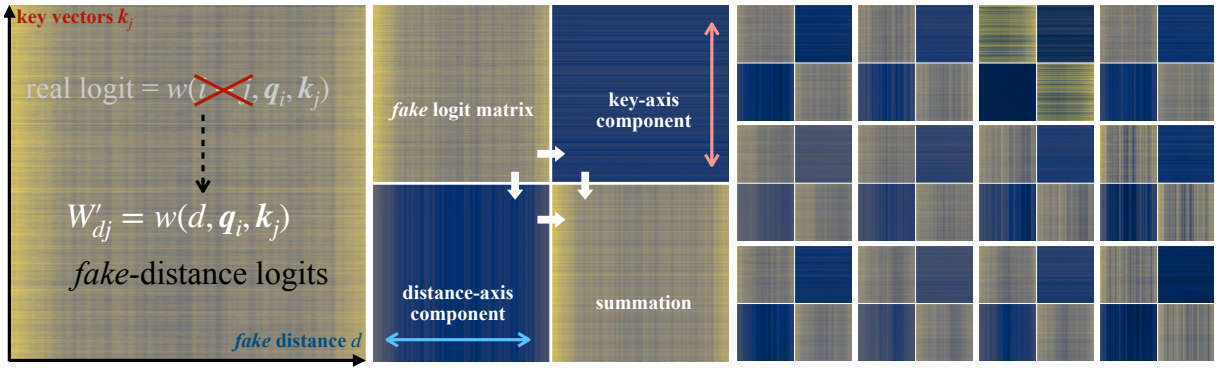
Inspired by this observation, we operate a ternary linear approximation of the logit map along the three axes. In other words, we examine if the logit map can be approximated with

$$W_{i,j} \approx a_{i-j} + b_i + c_j \tag{2}$$

with three arrays (or three linear components) of variables $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$. To obtain an approximation, we formulate this as a ridge regression, with more details in Appendix A. An example set of solved arrays is illustrated in Fig 2(c). [1]

By summing the three obtained components, the reconstructed logit matrix in Fig 2(d) shows striking similarity with the original matrix. With more

[1] Note that this is different from low-rank approximation of matrices, which approximates the full matrix (instead of the lower-trangular part), and only involves the row and column axes without diagonal components.



(a) logits with *fake* distances **(b) approximated with vector addition** **(c) more examples**

Figure 3: After replacing the distance value $i-j$ with a controlled fake distance d (illustrated in **(a)**), we find that a distance-axis + key-axis decomposition closely resembles the logit calculation. **(b)** illustrates the disentanglement process. The key-axis and distance-axis components align well with the patterns of the *fake*-distance logit matrix and sum up to a close approximation at the lower right corner. **(c)** presents additional examples, showing the prevalent applicability of such approximation. More details are provided in Sec. 3.2.

comparative examples shown in Fig 2(e) and Appendix D, we show the prevalence of this trend across layers and attention heads. This simple approximation has a correlation coefficient of 0.8650. This shows that a great majority of W's variance can be explained by Eq 2. Interestingly, its linear nature implies the logit map contains simple components that vary only depending on key, query, or position information individually, but not their combinations.

This approximation, however, only serves as a starting point for our study, as the $\boldsymbol{a}$ component assumes a static and global positional pattern a_{i-j} depending on token distance $i-j$, due to the course granularity of the analysis. We will move to a more fine-grained analysis in the next subsection.

3.2 The Disentanglement Law of Attention Logits

The previous section identified independent linear patterns in the logit map. However, in the calculation of the attention logits $w(i-j, \boldsymbol{q}_i, \boldsymbol{k}_j)$, the three variables still depend on each other. This prevents us from studying their effects on the logits individually. Additionally, the "query axis" does not have an actual effect on LLMs. It applies a uniform offset on each logit row, which is also a uniform offset in Eq 1. However, the softmax operator is invariant under uniform offsets. [2] So, it would make more sense to control the query axis while fully disentangling the effect of the position and semantics axes.

Therefore, instead of studying the real attention logits, we use a *fake* distance value d to replace the real distance $i-j$: $w(d, \boldsymbol{q}_i, \boldsymbol{k}_j)$. In light of the previous discussion, we fix a query vector $\boldsymbol{q}$ and visualize the following fake logit matrix $W' \in \mathbb{R}^{n\times n}$ where $W'_{d,j} = w(d, \boldsymbol{q}, \boldsymbol{k}_j)$ in Fig 3(a). After this substitution, the new matrix W' shows apparent vertical and horizontal patterns, suggesting prominent distance-wise and key-wise components.

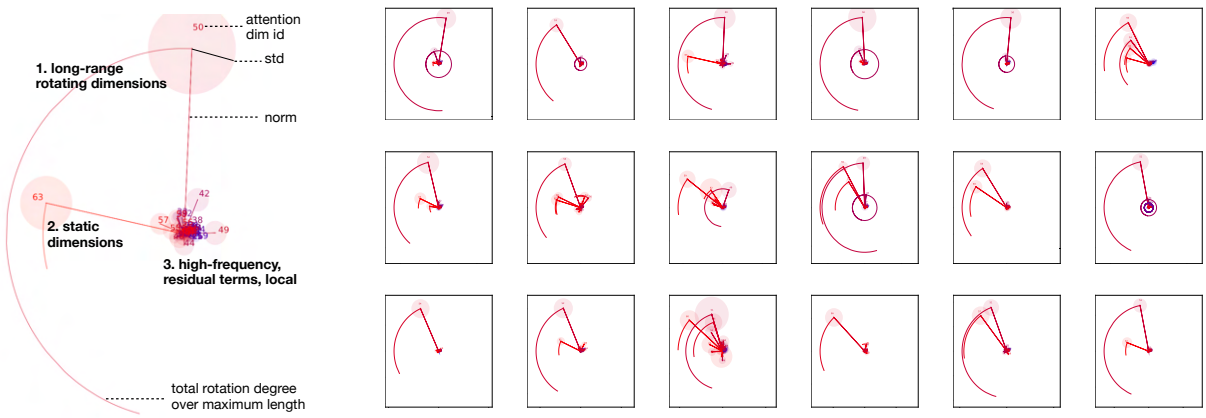
We follow on disentangling W' along these directions as $W'_{d,j} \approx a_d + b_j$.[3] The least-square ridge regression solution of this approximation has an explicit-form solution (with more details in Appendix B):

$$\begin{aligned} a_d &= \frac{1}{n}\sum_{j'} W'_{d,j'} - \frac{1}{2n^2}\sum_{d',j'} W'_{d',j'} \\ b_j &= \frac{1}{n}\sum_{d'} W'_{d',j} - \frac{1}{2n^2}\sum_{d',j'} W'_{d',j'} \end{aligned} \tag{3}$$

. Essentially, $\boldsymbol{a}$ and $\boldsymbol{b}$ are the average column and row of W', respectively, with a constant offset of $-\frac{1}{2n^2}\sum_{d',j'} W'_{d',j}$. This disentanglement process is visualized in Fig 3(b), where the key-axis and distance-axis components at two corners align with the patterns of the original *fake* logit matrix well. Once we combine these two components, their summation again demonstrates high similarity with the original matrix with details. This approximation has a correlation coefficient of 0.9470, explaining the vast majority of the logits' variance by the two simple 1-dimensional components. We list more

[2] The softmax weights remain the same after an offset: $\frac{\exp(w_i+c)}{\sum_j \exp(w_j+c)} = \frac{\exp w_i \cdot \exp c}{\sum_j \exp w_j \cdot \exp c} = \frac{\exp w_i}{\sum_j \exp w_j}$.

[3] This is a simplification of rank-2 matrix approximation.



(a) rotating pattern graph (b) more examples

Figure 4: Visualization of rotating query-key vector tuples in RoPE-based attention described in Section 3.3. **(a)** The rotating tuples averaged over tokens are plotted as arrows, with tuple indices annotated from 0 to $d/2-1$. Standard deviations over the tokens are shown as circles around endpoints, and the arc indicates the maximum rotation over the pre-training cutoff length. The sum of the tuples' projection along the horizontal axis is the actual logit value. **(b)** lists more of such figures, with more details in Sec 3.3

examples of such approximation in Fig 3(c) and Appendix D. These results indicate an approximated disentanglement of attention logits between positional relevance and semantic relevance:

$$w(i-j, \boldsymbol{q}, \boldsymbol{k}) \approx f(\boldsymbol{q}, i-j) + g(\boldsymbol{q}, \boldsymbol{k}) \quad (4)$$

. In other words, the majority of contribution from the position relation of two tokens $f(\boldsymbol{q}_i, i-j)$ is computed independently from their semantic relation $g(\boldsymbol{q}_i, \boldsymbol{k}_j)$ and added together.

3.3 The Mechanism in Query-Key Space

What caused the phenomena mentioned in the last two sections? Using the most prevalent positional encoding of RoPE as the subject of study, we delve deeper into the hidden features to show that certain feature dimensions of $\boldsymbol{q}$ and $\boldsymbol{k}$ are enforced with a large fixed norm and direction so that the approximation in Eq 4 is possible. Recall that the d-dimensional $\boldsymbol{q}$ and $\boldsymbol{k}$ are composed of a number of $d/2$ 2-tuples rotating at different angular speeds, with lower-indexed tuples rotating much faster than high-indexed ones. The overall logit

$$\begin{aligned} & w(i-j, \boldsymbol{q}, \boldsymbol{k}) \\ = & \sum_r \boldsymbol{k}_r^\top M_r^{rot}((i-j)\theta_r)\boldsymbol{q}_r \\ = & \sum_r \|\boldsymbol{k}_r\|\|\boldsymbol{q}_r\| \cos\left((i-j)\theta_r + \theta_{\boldsymbol{q}_r} - \theta_{\boldsymbol{k}_r}\right) \end{aligned}$$

, which are sums of cos values of rotating vectors with norms of $\|\boldsymbol{k}_r\|\|\boldsymbol{q}_r\|$, starting angle of $\theta_{\boldsymbol{q}_r} - \theta_{\boldsymbol{k}_r}$ and rotating speed θ_r per distance.

We plot how these vectors would rotate together on a 2-D plane in Fig 4(a). The starting positions of these rotating vectors are plotted as arrows pointing from the point of origin. Each arrow's tuple index ($\in \{0 \dots d/2-1\}$) is annotated beside the arrowheads. To visualize the randomness in these vectors, we also plot their standard deviation as circles around the endpoints. We also plot an arc to show the maximum rotation angle over the maximum distance allowed, i.e., the pre-training cutoff length $\theta_r^{\max} = \theta_r L_{\text{pre-train}}$. Notably, there exist a few (two in the shown example) *slow dominating* tuple dimensions with the following properties. We also provide asymptotic expressions of them for analysis later:

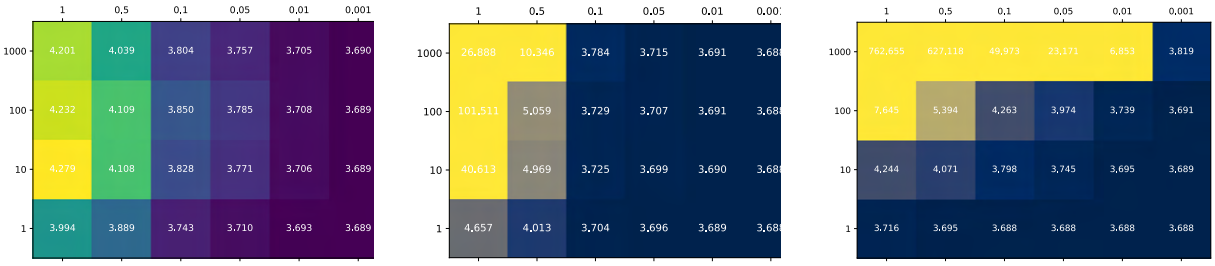
Observation 1. *Properties observed in slow-dominating features:*

1. *(Prominent dimensions) A relatively fixed average starting vector $\mathbb{E}\boldsymbol{k}_r$ with significantly larger norms than other dimensions.* $\exists \mathcal{R}_{slow}$ $\forall r' \notin \mathcal{R}_{slow}, r \in \mathcal{R}_{slow}, \|\boldsymbol{k}_{r'}\|\|\boldsymbol{q}_{r'}\| = o(\|\boldsymbol{k}_r\|\|\boldsymbol{q}_r\|)$. *Also* $\forall r \in \mathcal{R}_{slow}, \|\boldsymbol{k}_r - \mathbb{E}\boldsymbol{k}_r\| = o(\|\mathbb{E}\boldsymbol{k}_r\|)$.

2. *(Dimensions that are mostly static) The total rotation angle $\theta_r^{\max} = \theta_r L_{pre\text{-}train}$ is usually small if initial angle is close to π, i.e,* $\theta_r L_{pre\text{-}train} = o(\theta_{\boldsymbol{q}_r} - \theta_{\boldsymbol{k}_r} - \pi)$.

More similar patterns can be found in Fig 4(b) and Appendix D.

We theoretically demonstrate how these patterns account for the previous entanglement in the sense



(a) Effects of text transposition on LLM perplexity. x-axis controls the ratio of tokens perturbed, and y-axis controls the maximum distance of shuffled token pairs.

(b) Effects of feature transposition on LLM perplexity. x-axis controls ratio of tokens with position indices perturbed, and y-axis controls the maximum value position offset.

(c) Effects of position encoding manipulation on LLM perplexity. x-axis controls ratio of tokens with position indices perturbed, and y-axis controls the maximum value position offset.

Figure 5: Evaluating the impact of position information perturbation on LLMs' perplexity on ArXiv documents. With the vanilla perplexity being 3.688, our results show that shuffling text order in inputs and altering positional encodings in self-attention layers have limited effects on model perplexity and attention outputs.

that the contributions of *slow dominating* tuple dimensions to logits disentangle the positional and semantic components. *Other* tuple dimensions, however, contribute to relatively smaller variations in the logits. We have the following asymptotic disentanglement of the logit function (with proof in Appendix C):

Theorem 1. *If feature properties described in Observations 1 holds, then there exists functions $f(\boldsymbol{q}, i-j), g(\boldsymbol{q}, \boldsymbol{k})$ that so that the effect of $i-j$ and $\boldsymbol{k}$ can be asymptotically disentangled as:*

$$w(i-j, \boldsymbol{q}, \boldsymbol{k}) = (f(\boldsymbol{q}, i-j) + g(\boldsymbol{q}, \boldsymbol{k}))\,(1+o(1)) \tag{5}$$

.

The logit function is approximated as the sum of two functions f, g, which are only related to the positional and semantic relation between tokens. This provides computational explanations for the observations in the previous two sections. Not only are these functions existential, but the proof in Appendix C provides explicit-form solutions for f, g, which obtains a 0.959 linear correlation with the original logits. This further validates the observations in Section 3.1 and 3.2.

4 Position Generalization of LLMs

Taking insights of findings in Sec 3, this section explains how LLMs achieve position generalization towards *perturbed text positions* and *unseen lengths*. These phenomena reflect the aforementioned computational mechanism of disentangling position and semantics in attention: *positional relevance is not tightly bonded with semantic information in attention inference*. Instead, they contribute linearly independently to attention logits. In the following sections, we will empirically examine various forms of position generalization on the representation level.

Operation	**Qasper Accuracy**				
	0.5	0.1	0.05	0.01	0.001
Original	42.53				
Text Order	37.39	41.44	42.34	42.37	42.53
Feature Order	35.11	41.15	41.98	42.33	42.56
Position Encoding	37.19	42.44	42.42	42.74	42.64

Table 1: Effect of different levels of positional information perturbation on the Qasper Question-Answering dataset. Up to 5% of the tokens can be transposed or applied with perturbed position encoding (within token distance ± 5), while only resulting in a marginal effect on model accuracy.

4.1 Tolerance to Position Perturbations

Why can LLMs (like humans) read the language in shuffled word and sentence order? In light of the analysis in Sec 3, the positional information does not tightly bond with semantic relation but is more of an additive factor to the attention logit. We experimentally examine how this mechanism affects the capability of LLMs on various levels. At the superficial level, we show that transposing the positions of a ratio of words in a text sequence has marginal effects on model behaviors. To investigate the reason further at the representation level, we also mimic the effect of text word perturbations in the LLM representations, such as shuffling

the order of the feature sequences or modifying the position indices in positional encoding. More specifically:

1. randomly shuffling a ratio γ of the text orders in inputs within a maximum length $l_{\max}$ (**Text Transposition**),

2. randomly shuffling a ratio γ of the $\boldsymbol{k}$ feature order in each attention layer within a maximum length $l_{\max}$ (**Feature Order Transposition**),

3. randomly offsetting a ratio of γ of the $\boldsymbol{k}$'s position indices within self-attention layers within a range of $l_{\max}$ (**Position Encoding Manipulation**)

. We analyze the effects on attention output vectors and the corresponding LLMs' performance under these conditions.

The results, presented in Fig. 5, show that LLMs exhibit robustness to these perturbation methods. The original model has a perplexity of 3.688 on the ArXiv documents (Gao et al., 2020). Text transposition has a minimal impact on perplexity, with only a 0.02 increase when 1% of tokens are shuffled up to a distance of 1000 tokens. This suggests that LLMs do not rigidly depend on strict word order. Our intervention techniques simulate this linguistic transposition effect in Fig. 5b and 5c. Feature transposition also introduces a modest increase in perplexity, indicating that while position indices contribute to contextual representation, their precise ordering is not always critical in each self-attention layer. As further analysis, when the position encoding contains perturbed indices, the perplexity still has a marginal increase when 10% of token positions were perturbed by ± 10, or increase by an absolute value of 0.01 when 1% of tokens has position encoding perturbed by ± 10. These phenomena further align with the previous observations that position information acts as a disentangled additive factor rather than being tightly entangled with semantic relationships.

As perplexity might not reflect a model's actual performance on downstream tasks, we evaluate how Llama-3.2-3B-Instruct performs on the Qasper dataset (Dasigi et al., 2021) under these conditions. Results are listed in Table 1. Similar to the findings on the model perplexity, the model can tolerate 5% or word order being shuffled up to 5 token distance in the inputs, with only a 0.6% drop in accuracy. When we perturb the positional information inside the model, the model exhibits flexibility (<0.1% drop in accuracy) under both feature order transposition and position encoding manipulation when the position information of up to 10% of the features is perturbed.

4.2 When and How LLMs Generalize to Longer Texts

Recent techniques like LM-Infinite and InfLLM enable LLMs to generalize to longer text sequences than those encountered during training. The common practice adopted by these techniques is to modify the relative position before applying the original self-attention mechanism. This is equivalent to applying self-attention over a modified $[\boldsymbol{k}_i]$ and $[\boldsymbol{v}_i]$ sequence with probably, which might be different (sometimes significantly shorter) than the original $[\boldsymbol{k}_i]$ and $[\boldsymbol{v}_i]$ sequences. More specifically, in those techniques, the resulting sequence of features usually appears as if they are of the following positional distances:

$$[L_{PT}, \cdots, L_{PT}, L_{PT}-1, \cdots, l_L, \cdots, l_L, \cdots, 1, 0]$$

, where L_{PT} is the pre-training maximum length, and l_L is a position used technically for storing a few automatically retrieved feature vectors in the extremely long context. The retrieved features are usually used to enhance information retrieval.

This is in contrast to the intuition we obtained from common machine learning practices: why do LLMs train purely from shorter texts that generalize to extreme lengths (e.g., 200M in LM-Infinite) with only minor modifications to the model architecture? Moreover, little explicit design was implemented in modern SotA LLMs to enable this extreme generalization. This phenomenon could find support in our analysis: *even though posed to unseen extreme lengths, LLMs do not bind positional relevance information with the semantic features of the contextual tokens.* In other words, the $\boldsymbol{k}$ and $\boldsymbol{v}$ vectors could be interpreted as a pool of semantic features. The self-attention mechanism approximately and additively applies the position component to the pool. As long as the resulting distance sequence is similar to its normal shape $[L_{PT}, L_{PT}-1, \ldots, 1, 0]$, the attention output vector will reside in its normal distribution. Therefore, technically, the attention output vector is still in-distribution, so the remaining parts of LLMs on top of the attention outputs will not take out-of-distribution features as inputs.

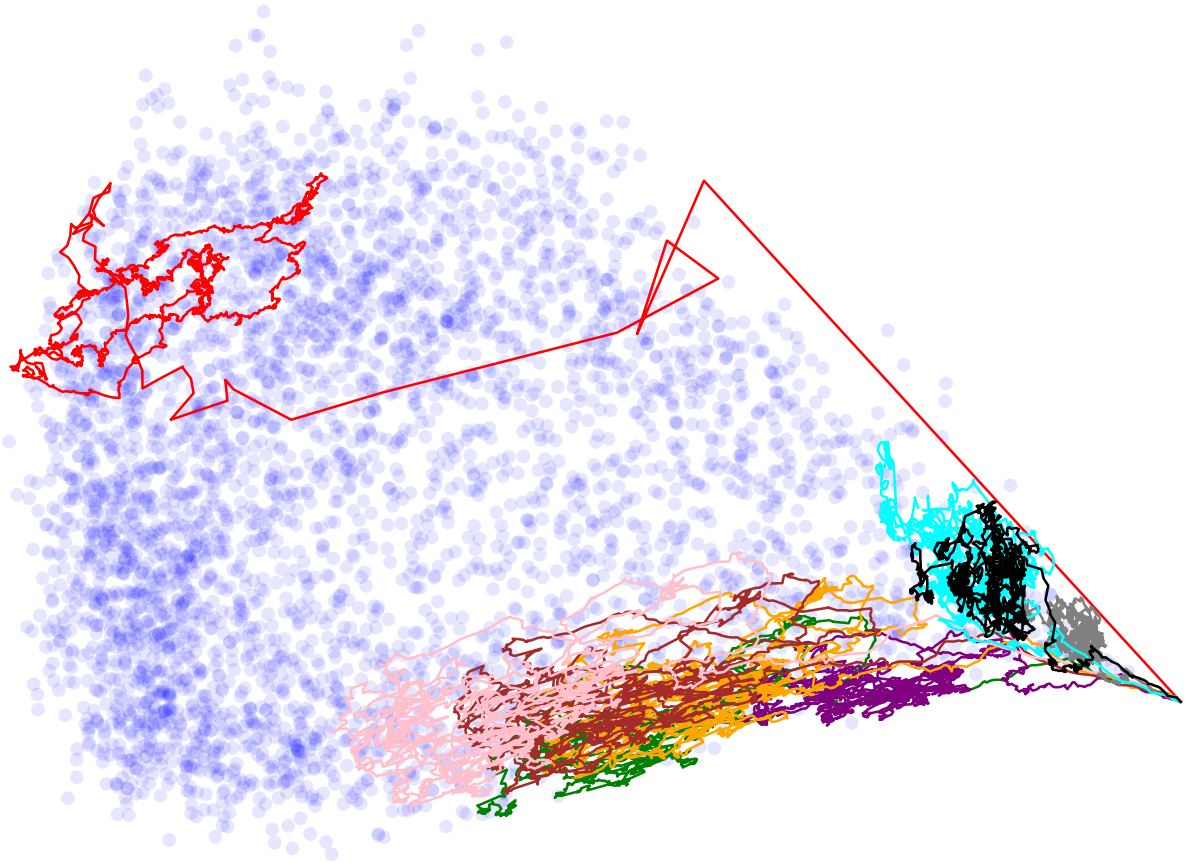


Figure 6: Visualization of attention output vectors projected onto a 2-D plane using PCA. Colored broken lines trace attention output vectors $\boldsymbol{o}$ across a sliding window of key and value vectors in length extension techniques. This shows that the output vectors remain within the normal distribution, supporting our explanation of the possibility of length generalization.

To verify this claim, we visualize the attention output vectors of an arbitrary layer using the technique above. This is a projection down to a 2-D plane using PCA[4]. The blue dots are the normal attention output vectors, which mark their normal distribution. Then, we select a set of $\boldsymbol{q}$ vectors and associate them with different colors. For each vector, we apply it over a subsequence of length L_{PT}: $[\boldsymbol{k}_i, \boldsymbol{k}_{i+1}, \ldots, \boldsymbol{k}_{i+L_{PT}}]$ and $[\boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \ldots, \boldsymbol{v}_{i+L_{PT}}]$. As we vary the value of starting position i, we trace the output vector with colored broken lines. As shown in Figure 6, these lines, though extending to different directions and different ranges depending on the $\boldsymbol{q}$, still wander within the range of normal attention output vector distribution. This further validates our explanation of the length generalization and provides insights for future manipulation of the self-attention module for research purposes.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

5 Conclusions and Future Work

In this work, we investigated the computational mechanisms behind the position generalization capabilities of LLMs. We first demonstrated that attention logits in LLMs can be approximately disentangled into independent components representing positional and semantic relevance. This finding suggests a structured decomposition within the model's internal computations. Through empirical analysis, we further examined various forms of position generalization at the LLM representation level. These insights provide both computational explanations and insights into controlling these phenomena.

Future research could delve deeper into the specific architectural choices and training data patterns that contribute to this robustness. Additionally, this work serves only as a starting point in mechanically analyzing position processing.

Limitations

While our study provides new insights into the computational mechanisms behind position generalization in LLMs, several limitations remain. First, our study primarily evaluates position robustness involving text order and length generalizations. While these are valuable computational linguistic phenomena, real-world language processing tasks often involve more complex positional dependencies, such as discourse coherence, document-level reasoning, and hierarchical structures. Future work could explore much more complicated scenarios. Second, our findings suggest that position and semantic components of attention logits can be disentangled, but the extent to which models actively leverage this property during training is unclear. Future explanations on how such a mechanism is acquired during training dynamics could greatly enhance the work.

References

Dik Bakker. 1998. Flexibility and consistency in word order patterns in the languages of europe. *Empirical Approaches to Language Typology*, 20:383–420.

Jerome S Bruner and Donald O'Dowd. 1958. A note on the informativeness of parts of words. *Language and Speech*, 1(2):98–101.

Manuel Carreiras, Ileana Quiñones, Juan Andrés Hernández-Cabrera, and Jon Andoni Duñabeitia. 2015. Orthographic coding: brain activation for letters, symbols, and digits. *Cerebral Cortex*, 25(12):4748–4760.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jon Andoni Duñabeitia, Maria Dimitropoulou, Jonathan Grainger, Juan Andrés Hernández, and Manuel Carreiras. 2012. Differential sensitivity of letters, numbers, and symbols to character transpositions. *Journal of Cognitive Neuroscience*, 24(7):1610–1624.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Javier Garcia-Orza, Manuel Perea, and Samara Munoz. 2010. Are transposition effects specific to letters? *Quarterly Journal of Experimental Psychology*, 63(8):1603–1618.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008.

Sylvain Kahane, Ziqian Peng, and Kim Gerdes. 2023. Word order flexibility: a typometric study. In *Depling, GURT/SyntaxFest 2023*. Association for Computational Linguistics (ACL).

Elsi Kaiser and John C Trueswell. 2004. The role of discourse context in the processing of a flexible word-order language. *Cognition*, 94(2):113–147.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Doris L Payne. 1992. *Pragmatics of Word Order Flexibility*, volume 22. John Benjamins Publishing.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160.

Ofir Press, Noah Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Graham Rawlinson. 2007. The significance of letter position in word recognition. *IEEE Aerospace and Electronic Systems Magazine*, 22(1):26–27.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. 2024. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021a. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. Unnatural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Benedikt Szmrecsanyi. 2016. An informationtheoretic approach to assess linguistic complexity. *Complexity, isolation, and variation*, 57:71.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

A Details of Solving Ternary Linear Approximation in Sec 3.1

The total residue square in Eq 2 is represented as:

$$\mathcal{L}(W; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \sum_{j \le i} (W_{i,j} - a_{i-j} - b_i - c_j)^2 \quad (6)$$

. This is a strictly convex function, so one single optimal solution exists. At the optimum, the objective has zero gradient $\nabla \mathcal{L}(W; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = \mathbf{0}$. Taking derivative over all variables, this requirement is equivalent to a linear system:

$$\begin{aligned} (n-i)a_i + \sum_{j \le n-i} b_j + \sum_{j \ge i} c_j &= \sum_{j \ge i} W_{i+j,j} \\ \sum_{j \le n-i} a_j + (n-i)b_i + \sum_{j \ge i} c_j &= \sum_{j \ge i} W_{j,i} \\ \sum_{j \le i} (a_j + b_j) + (i+1)c_i &= \sum_{j \le i} W_{i,j} \end{aligned} \quad (7)$$

. We then apply a linear equation solver[5] to this system.

B Details of the Approximation in Sec 3.2

The total residue square in Eq 2 is represented as:

$$\mathcal{L}(W'; \boldsymbol{a}, \boldsymbol{b}) = \sum_{d,j} (W'_{d,j} - a_d - b_j)^2 \quad (8)$$

. Taking derivative over all variables, the optimal point satisfies a linear system:

$$\begin{aligned} n a_d + \sum_{j'} b_{j'} &= \sum_{j'} W'_{d,j'} \\ \sum_{d'} a_{d'} + n b_j &= \sum_{d'} W'_{d',j} \end{aligned} \quad (9)$$

. The solution set is the following family of values, where c can take arbitrary values.

$$\begin{aligned} a_d &= \frac{1}{n} \sum_{j'} W'_{d,j'} + c - \frac{1}{n^2} \sum_{d',j'} W'_{d',j'} \\ b_j &= \frac{1}{n} \sum_{d'} W'_{d',j} - c \end{aligned} \quad (10)$$

. Adding an l_2-norm regularization term with any weight, as c is the only free variable here, the optimal solution will be the point where.

$$\sum_d a_d - \sum_j b_j = 0 \quad (11)$$

[5] Adopted solver in NumPy: https://numpy.org/doc/2.2/reference/generated/numpy.linalg.solve.html. We add a 1e-6 l_2-norm regularization for numerical stability

. That will require $c = \frac{1}{2n^2} \sum_{d',j'} W'_{d',j'}$. The final solution will become:

$$\begin{aligned} a_d &= \frac{1}{n} \sum_{j'} W'_{d,j'} - \frac{1}{2n^2} \sum_{d',j'} W'_{d',j'} \\ b_j &= \frac{1}{n} \sum_{d'} W'_{d',j} - \frac{1}{2n^2} \sum_{d',j'} W'_{d',j'} \end{aligned} \quad (12)$$

.

C Asymptotic Disentanglement of Attention Logit Function

Proof. In those slow-dominating dimensions $r \in \mathcal{R}_{\text{slow}}$, denote $\theta_\delta = \theta_{\boldsymbol{q}_r} - \theta_{\boldsymbol{k}_r} - \pi$ and $\bar{\boldsymbol{k}} = \mathbb{E}\boldsymbol{k}$. We have:

$$\begin{aligned} & \boldsymbol{k}_r^\top M_r^{rot}((i-j)\theta_r)\boldsymbol{q}_r \\ =& \|\boldsymbol{k}_r\| \|\boldsymbol{q}_r\| \cos(\theta_{\boldsymbol{q}_r} - \theta_{\boldsymbol{k}_r} + (i-j)\theta_r) \\ =& \|\bar{\boldsymbol{k}_r} + (\boldsymbol{k}_r - \bar{\boldsymbol{k}_r})\| \|\boldsymbol{q}_r\| \\ & \cos(\pi + \theta_\delta + (i-j)\theta_r) \\ \le& -(\|\bar{\boldsymbol{k}_r}\| + \|\boldsymbol{k}_r - \bar{\boldsymbol{k}_r}\|)\|\boldsymbol{q}_r\| \\ & \cos(\theta_\delta + (i-j)\theta_r) \\ =& -\|\bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos(\theta_\delta + (i-j)\theta_r) \\ & - \|\boldsymbol{k}_r - \bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \\ & \left(\cos\theta_\delta - \sin\left(\theta_\delta + \frac{1}{2}(i-j)\theta_r\right) \sin\frac{1}{2}(i-j)\theta_r\right) \\ =& -(\|\bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos(\theta_\delta + (i-j)\theta_r) \\ & + \|\boldsymbol{k}_r - \bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos\theta_\delta)(1 + o(1)) \end{aligned}$$

□

Then, the faster-rotating dimensions have contributions smaller than slower ones:

$$\begin{aligned} & \sum_{r \notin \mathcal{R}_{\text{slow}}} \boldsymbol{k}_r^\top M_r^{rot}((i-j)\theta_r)\boldsymbol{q}_r) \\ (r_0 \in \mathcal{R}_{\text{slow}}) =& |\mathcal{R}| o(\|\boldsymbol{k}_{r_0}\| \|\boldsymbol{q}_{r_0}\|) \\ =& o\left(\sum_{r \in \mathcal{R}_{\text{slow}}} \boldsymbol{k}_r^\top M_r^{rot}((i-j)\theta_r)\boldsymbol{q}_r\right) \end{aligned} \quad (13)$$

In summary:

$$\begin{aligned} & w(i-j, \boldsymbol{q}, \boldsymbol{k}) \\ =& -\sum_{r \in \mathcal{R}_{\text{slow}}} (\|\bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos(\theta_\delta + (i-j)\theta_r) \\ & + \|\boldsymbol{k}_r - \bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos\theta_\delta)(1 + o(1)) \\ =& (f(\boldsymbol{q}, i-j) + g(\boldsymbol{q}, \boldsymbol{k}))(1 + o(1)) \end{aligned} \quad (14)$$

if we define

$$
\begin{aligned}
f(\boldsymbol{q}, i-j) &= -\sum_{r \in \mathcal{R}_{\text{slow}}} \|\bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos(\theta_\delta + (i-j)\theta_r) \\
g(\boldsymbol{q}, \boldsymbol{k}) &= -\sum_{r \in \mathcal{R}_{\text{slow}}} \|\boldsymbol{k}_r - \bar{\boldsymbol{k}_r}\| \|\boldsymbol{q}_r\| \cos \theta_\delta
\end{aligned}
\tag{15}
$$

, respectively.

D More Example Visualization from Other Models

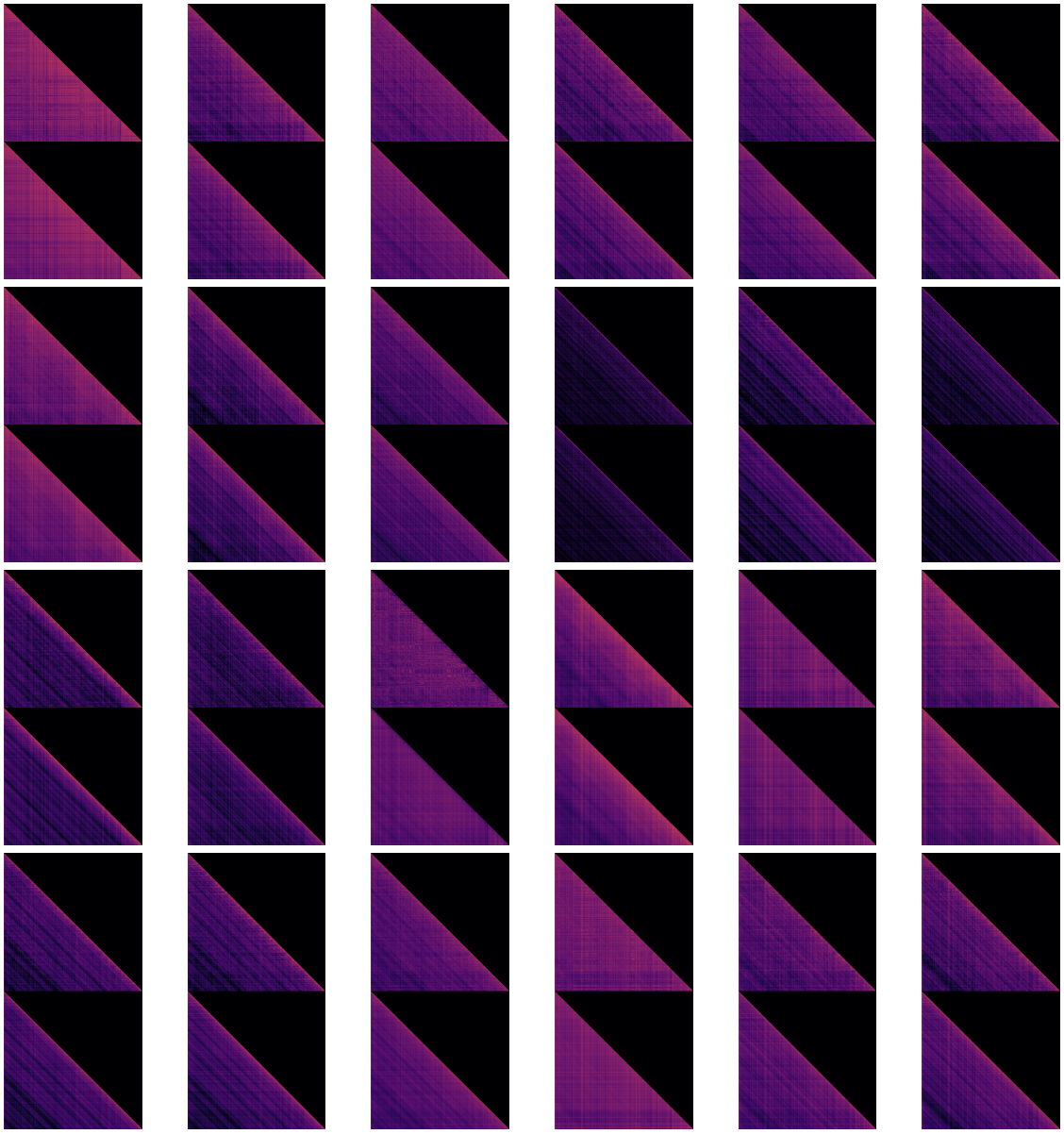


Figure 7: More examples of 3-axis approximation of logit matrix on Llama-2 model.

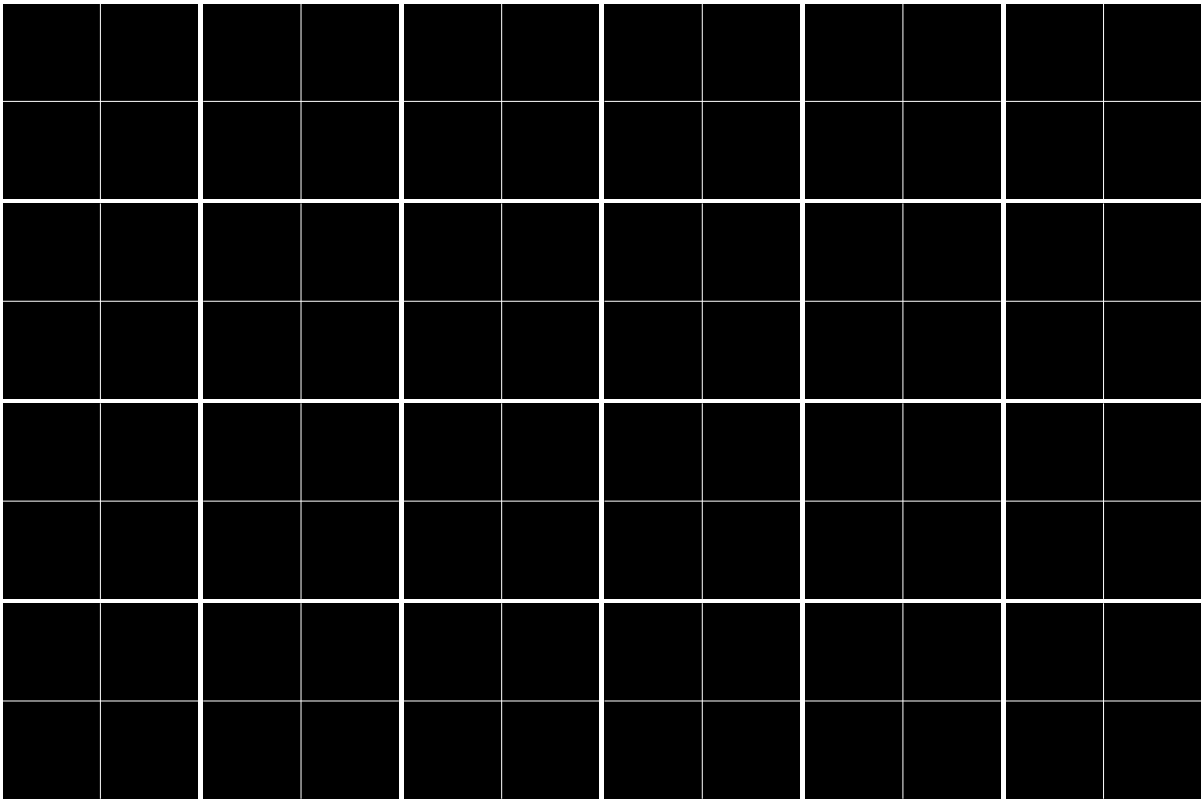
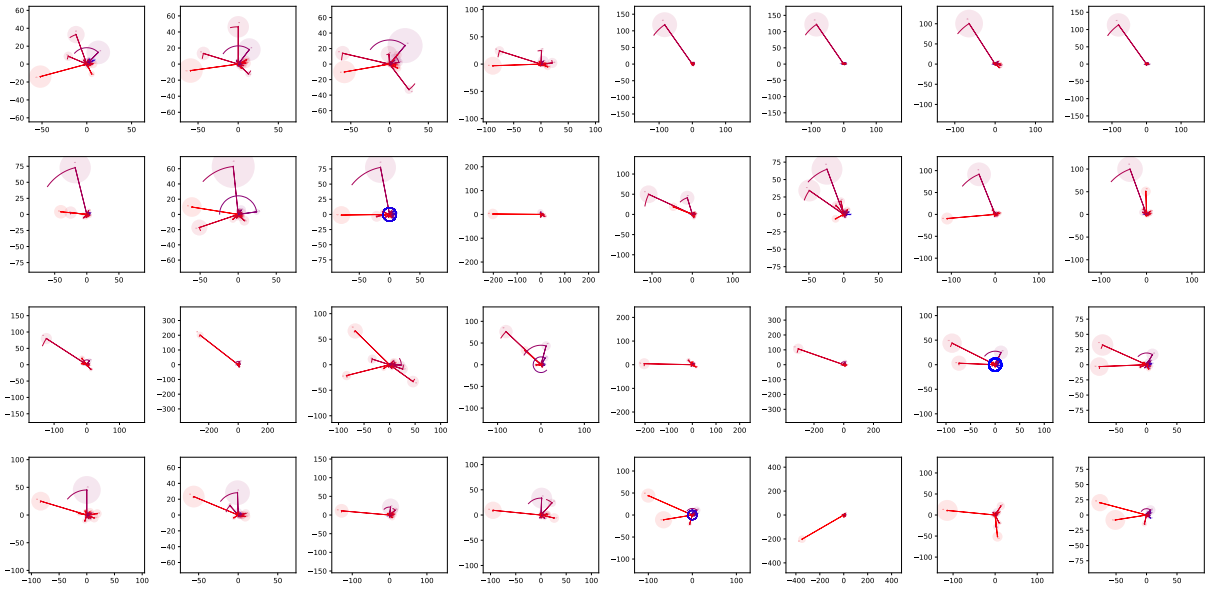
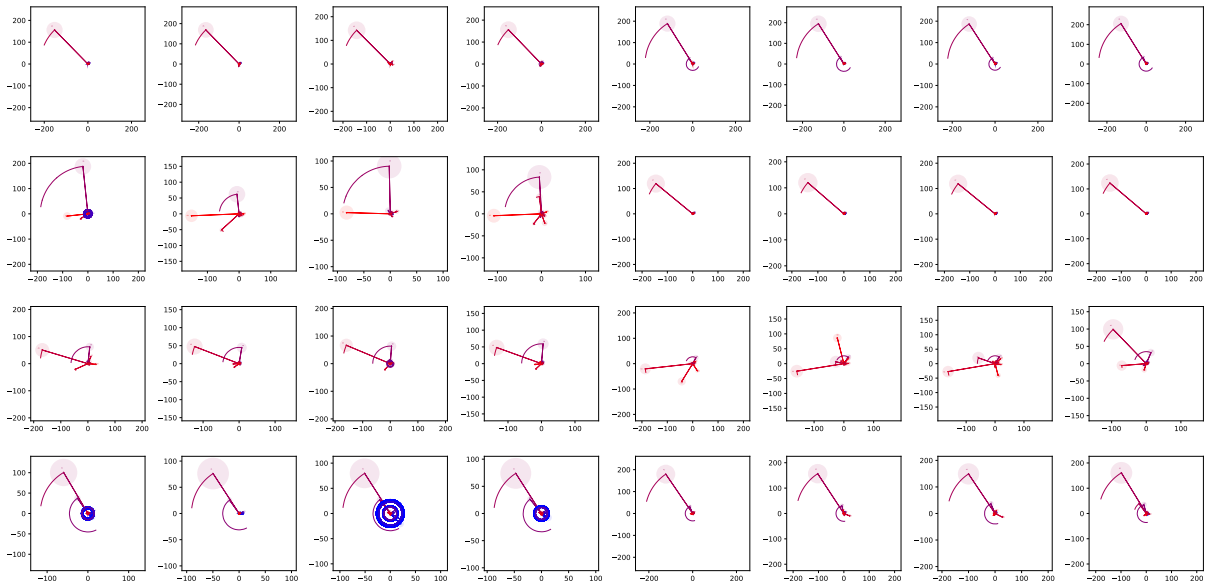


Figure 8: More examples of disentanglement of fake-distance logit matrix on Llama-2 model.



(a) Llama-2.



(b) Llama-3

Figure 9: More examples of the rotating vector tuples in RoPE-based attention in other models.