High-Order Self-Attention Mechanism: A Deep Attention Model in Extended Parameter Space

Anonymous ACL submission

Abstract

Since the introduction of self-attention in 2016, Transformer based pre-training model have achieved remarkable success, driving breakthroughs across various NLP tasks. Inspired by graph attention node aggregations from neighbor nodes, we revisit the self-attention mechanism to explore its potential for capturing higher-order relationships in sequence modeling. Specifically, we propose a novel High-Order Self-Attention mechanism, which enhances the expressive power of traditional self-attention through multiple self-attention aggregations and positional embeddings. By integrating this mechanism into self-attention based models during the pre-training process with limited data and model capacity, we achieve up to a 35% improvement in accuracy for RoBERTa on masked token prediction tasks and up to a 75% increase in ROUGE-2 scores for GPT-2 on pre-training task, under identical experimental conditions demonstrating the robustness and efficiency of the proposed method. This mechanism further enables a novel parameter stacking approach, allowing models to achieve more efficient and scalable training. These findings demonstrate the potential of High-Order Self-Attention for advancing sequence modeling and pre-training workflows.

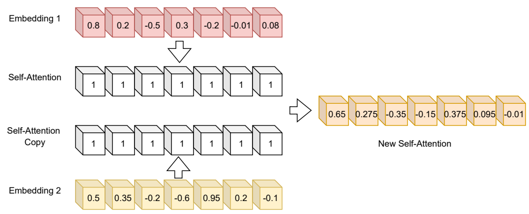

Figure 1: High-Order Self-Attention Mechanism: a standard self-attention computation is broadcast-multiplied with High-Order Position Embeddings (HOPE) to generate multiple diverse attention units, each capturing different dependency patterns. Guided by HOPE, the standard self-attention outputs are aggregated through a feed-forward network (FFN) to produce a new attention representation. This aggregated representation forms high-order relationships with the original attention, thereby enhancing the model's ability to distinguish relevant information.

1 Introduction

The self-attention mechanism has revolutionized sequence modeling by providing a dynamic approach to capturing global dependencies across data. By allowing each element in a sequence to evaluate its relationships with all other elements, self-attention enables the extraction of context-aware representations through dot-product similarity. This ability to capture both short-range and long-range dependencies makes self-attention a cornerstone of modern natural language processing (NLP).

This transformative capability is exemplified by Transformer-based architectures such as BERT (?) and GPT-2 (Radford et al., 2019), which have demonstrated exceptional performance across a wide range of NLP tasks, including machine translation, question answering, and text generation. These models excel at processing long and complex texts, firmly establishing self-attention as the foundation of large language models (LLMs).

However, traditional self-attention mechanisms exhibit significant limitations when dealing with the intricacies of natural language. Natural language is inherently hierarchical, comprising layers of meaning that span from local dependencies (e.g., between words) to global semantics (e.g., paragraph-level relationships). Existing methods primarily rely on simple dot-product calculations, which, while effective for capturing immediate dependencies, struggle to represent multi-level, hierarchical relationships within sequences.

To address these challenges, the concept of high-order dependencies offers a promising direction. High-order dependencies extend beyond direct relationships by capturing more nuanced and multi-level interactions within sequences. Previous works have explored high-order attention in specific tasks such as person re-identification (Chen et al., 2019), remote sensing (Zhang et al., 2020), and graph representation learning (Veličković et al., 2018), where attention is applied over relational structures. However, these designs are either task-dependent or limited to non-sequential data, and fundamentally differ from our approach. Introducing mechanisms capable of modeling such dependencies is critical for improving both the expressiveness and performance of attention-based architectures.

Inspired by these challenges, and partially motivated by insights from Graph Attention Networks (GAT) (Veličković et al., 2018), we propose a novel High-Order Self-Attention Network (HOSA) to overcome the limitations of traditional self-attention, shown in Figure 1. HOSA enhances the self-attention framework by incorporating high-order dependency modeling through multi-level attention aggregation. This is achieved by introducing multiple replicated attention heads, each modulated by unique positional embeddings, and linearly aggregating their contributions.

The main contributions of this work can be summarized as follows:

We introduce a novel self-attention mechanism that explicitly incorporates high-order dependency modeling, significantly enhancing the ability to capture multi-level relationships in sequences. By integrating HOSA into pre-trained architectures like GPT-2 and RoBERTa, we achieve significant gains in masked and next-token prediction tasks. Extensive experiments on benchmarks show HOSA consistently surpasses traditional self-attention.

2 Related Work

Various attention variants have been proposed to improve the efficiency or inductive bias of self-attention, such as linear attention (Katharopoulos et al., 2020), light attention (Vaswani et al., 2021), and adaptive sparse attention (Beltagy et al., 2020). While our work is inspired by the need to overcome the limitations of standard dot-product attention, our proposed mechanism is structurally independent from these designs and focuses instead on modeling multi-level token dependencies through a high-order formulation.

2.1 Self-Attention Mechanism

The self-attention mechanism was significantly advanced and popularized through the Transformer architecture by Vaswani et al. (Vaswani et al., 2017), revolutionizing sequence modeling by enabling models to dynamically compute dependencies between all elements in a sequence. This global attention mechanism allows the model to effectively capture contextual information, making it the cornerstone of state-of-the-art architectures like BERT and GPT. The mathematical formulation of single-layer, single-head self-attention is as follows:

$$e = \frac{\mathbf{K}\mathbf{Q}^T}{\sqrt{d_k}} \tag{1}$$

$$\alpha = softmax(e) \tag{2}$$

$$\vec{h}' = \sigma(\alpha \mathbf{V}) \tag{3}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query, key, and value matrices obtained by applying linear transformations to the input sequence. The scaling factor $\sqrt{d_k}$ ensures numerical stability for larger dimensions d_k. By assigning importance weights α to elements in the sequence, this mechanism enables the model to capture global dependencies effectively. Despite its success, standard self-attention mechanisms often struggle to capture higher-order relationships or hierarchical dependencies, which are crucial for tasks requiring a deeper understanding of data structures.

2.2 Self-Attention in BERT and the Role of Positional Embeddings

BERT builds on the Transformer architecture by introducing bidirectional self-attention, allowing the model to utilize both preceding and succeeding context for tasks such as masked token prediction. However, in masked token prediction, the bidirectional nature of self-attention introduces a fundamental limitation: when masked tokens share identical contextual embeddings, the attention mechanism cannot inherently distinguish them. Without additional information, attention weights are identical, rendering the model unable to disambiguate these tokens.

To address this, BERT incorporates positional embeddings as an essential component of the model. These embeddings encode the position of each token in the sequence, enabling the attention mechanism to differentiate tokens based on their positions. While effective, standard positional embeddings are fixed and static, limiting their ability to capture hierarchical or high-order positional relationships, which are crucial for tasks involving complex token interactions.

Building on this concept, our work introduces High-Order Positional Embedding (HOPE) as a critical component of the HOSA framework. HOPE extends the role of positional embeddings by directly integrating positional information into the attention mechanism, allowing for more nuanced token differentiation and the modeling of hierarchical relationships.

2.3 Applications of Self-Attention in GPT

GPT adopts a unidirectional self-attention mechanism, where each token attends only to its preceding tokens in the sequence. This design aligns well with autoregressive tasks such as text generation, enabling GPT to predict the next token based on the preceding context. While GPT does not encounter issues related to bidirectional attention, it still relies on positional embeddings to encode sequence order.

In this study, we evaluate HOSA and its positional embedding component (HOPE) within GPT's architecture to assess their effectiveness in autoregressive tasks. These experiments demonstrate the versatility of HOSA across different sequence modeling paradigms and validate HOPE's potential for enhancing positional information handling in various contexts.

3 Methodology

In this section, we present the High-Order Self-Attention Network (HOSA), a novel architecture designed to enhance the representational capacity of self-attention mechanisms through the integration of multiple parallel self-attention units within each HOSA module. HOSA leverages BERT's token position embeddings and incorporates High-Order Position Embedding (HOPE) to differentiate between multiple self-attentions, enabling effective token representation learning. The final hidden states computed by HOSA capture complex relationships within the sequence.

Key notations used in this section: $batch\_size$: the batch size during training. max_len: the maximum length of input sentences. $hidden\_dim$: the dimensionality of hidden feature representations. num_hosa: the number of attention units in HOSA.

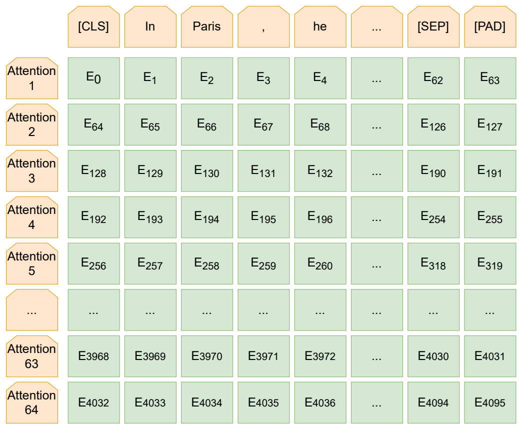

Figure 2: HOSA utilizes High-Order Position Embedding (HOPE) for its inputs. For example, with an input sentence of length 64 and number of attention units 64, HOPE produces a 64×64 positional embedding matrix that encodes pairwise token positions. In BERT, E_x denotes the position embedding of token x, whereas in HOSA, $E_{i,j}$ spans the entire $num\_hosa \times max\_len$ space, capturing relationships across multiple self-attention computations.

3.1 High-Order Position Embedding (HOPE)

BERT uses token position embeddings to distinguish masked tokens during pre-training. Building on this concept, HOSA introduces the High-Order Position Embedding (HOPE), denoted as E_{hope}, to serve as a unique identifier for multiple standard self-attention layers within the network. As illustrated in Figure 2, HOPE generalizes position embeddings to provide distinct representations for different self-attention mechanisms, ensuring they are uniquely distinguished during computation.

HOPE is represented as a tensor of shape $\mathbb{R}^{\text{num\_hosa}\times\text{max\_len}\times\text{hidden\_dim}}$, where each element serves as a unique identifier for different attention computations. In HOSA, HOPE is first applied to the key matrix $\mathbf{K}$ via element-wise multiplication using tensor broadcasting, resulting in the extended key matrix $\mathbf{K}_{hope}$. The matrix $\mathbf{K}_{hope}$ is then used in the attention mechanism, where it is multiplied with the query matrix $\mathbf{Q}$. Unlike the traditional positional embeddings in BERT, which are added to token embeddings, HOPE directly interacts with the attention mechanism, enhancing its ability to differentiate between attention computations. Mathematically, the extension of the key matrix can be expressed as:

$$\mathbf{K}_{hope} = \mathbf{K} \odot \mathbf{E}_{hope} \tag{4}$$

where $\mathbf{Q}$ represents the query matrix, $\mathbf{K}$ is the key

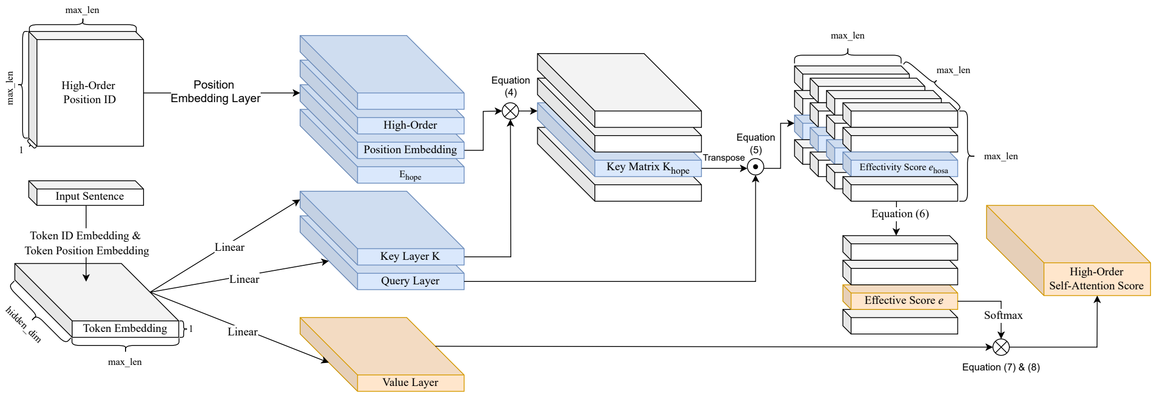

Figure 3: Model architecture of the High-Order Self-Attention Mechanism. The blue-shaded section represents the High-Order computation, whereas the yellow-shaded section illustrates the sequence computation in standard Self-Attention. The symbol $\otimes$ represents the inner product computation with broadcasting, while $\odot$ denotes element-wise multiplication. The equation numbers correspond to the equation in the Methodology section. The multi-layer modules clearly illustrate how dimensions are selected and multiplied across different computational steps.

matrix, $\mathbf{E}_{hope}$ is the High-Order Position Embedding, and $\odot$ denotes element-wise multiplication.

3.2 High-Order Self-Attention Mechanism

The High-Order Self-Attention Mechanism (HOSA) is designed to enhance the traditional self-attention mechanism by incorporating high-order dependencies and positional information, illustrated in Figure 3. By leveraging High-Order Position Embeddings (HOPE), HOSA emphasizes the model's ability to learn specific patterns, effectively reducing attention to irrelevant tokens while enhancing attention to relevant ones. This enhancement primarily reflects the model's capacity to fit underlying dependency patterns in the data, rather than directly capturing semantic relationships. Specifically, HOPE is designed as a mathematical mechanism to model structural information in the sequence, enabling the model to more efficiently capture rich contextual dependencies when significant patterns exist in the data.

High-Dimensional Attention Score Calculation HOSA first computes a high-dimensional attention score e_{hosa} to capture high-order relationships:

$$e_{\text{hosa}} = \frac{\mathbf{K}_{hope} \cdot \mathbf{Q}^T}{\sqrt{d_k}} \tag{5}$$

Here, $\mathbf{K}_{hope}$ is the extended key matrix derived from HOPE, and $\mathbf{Q}$ is the query matrix. The division by $\sqrt{d_k}$ ensures numerical stability and scales the attention scores appropriately. This step encodes high-order dependencies by incorporating rich positional information from $\mathbf{K}_{hope}$.

Dimensionality Reduction via Projection To reduce the additional dimension introduced by $\mathbf{K}_{hope}$, HOSA applies a learnable projection matrix $\mathbf{W} \in \mathbb{R}^{\text{num\_hosa} \times \text{max\_len} \times 1}$:

$$e = e_{\text{hosa}} \cdot \mathbf{W} \tag{6}$$

This operation collapses the redundant dimension while preserving the most salient patterns, enabling the model to focus on important token dependencies.

Attention Weight Normalization The reduced attention scores e are then normalized using the Softmax function:

$$\alpha = \text{softmax}(e) \tag{7}$$

The Softmax operation ensures that attention weights are distributed over the tokens, amplifying relevant tokens while suppressing irrelevant ones.

Final Representation Update The normalized attention weights α are used to compute the final representation:

$$\mathbf{h}' = \sigma(\alpha \mathbf{V}) \tag{8}$$

Here, $\mathbf{V}$ represents the value matrix, and σ is an optional activation function that introduces non-linearity. This step produces the updated token representation, enriched with contextual and hierarchical information.

The computational complexity of our model is determined to be $O(n^2)$, where n is the maximum sentence length.

4 Experiment

4.1 Experiment Design

To evaluate the impact of the High-Order Self-Attention Mechanism (HOSA) in self-attention-based models, we selected two compact yet classic pre-trained models, GPT-2 and RoBERTa, for comparison. These models serve as representative benchmarks for assessing the effectiveness of HOSA in enhancing self-attention mechanisms.

GPT-2 is chosen as it serves as the foundation for modern large language models (LLMs), leveraging unidirectional self-attention. In contrast, RoBERTa represents a bidirectional self-attention architecture. Both models have achieved state-of-the-art performance on various natural language processing tasks, making them ideal reference points for this study. The dataset used for the experiments is the BookCorpus dataset, which is widely used for pre-training language models, including BERT.

Two experimental tasks were designed:

- **Next Token Prediction**: Evaluates the model's ability to predict the next token in a sequence, simulating a generative setting.
- **Masked Token Prediction**: Assesses the model's capability to predict randomly masked tokens, reflecting a masked language model task.

Key experimental parameters include embedding dimensions, sentence lengths, the number of layers, and the number of attention heads, ensuring a controlled and comprehensive evaluation.

4.2 Dataset and Training Strategy

To evaluate HOSA in Experiments 1 and 2, a scaled dataset of 100,000 samples was selected from the BookCorpus benchmark (Zhu et al., 2015). This choice balances computational efficiency and scalability while maintaining sufficient data diversity for reliable evaluation.

The BookCorpus dataset has an average sentence length of 15.7 tokens, with 99.87% of sentences below 64 tokens, making it a suitable benchmark for testing HOSA's performance. Training was conducted over 60 epochs using a linear descent learning strategy, with a consistent batch size of 50 across all experiments to ensure comparability.

4.3 Experiment 1: Validation of HOSA in Next Token Prediction

The primary objective of Experiment 1 is to evaluate the effectiveness of HOSA in next token prediction tasks, a generative pre-training scenario commonly associated with GPT-2. By replacing the scaled dot-product self-attention module in GPT-2 with HOSA, we assessed its performance under various configurations to explore its robustness and generalizability.

The experimental setup covered a range of key configurations, including embedding dimensions, number of layers, sentence lengths, and attention heads. Specifically, the embedding dimensions ranged from 128 to 3200, with the number of layers varying from 1 to 16. Sentence lengths were tested at 64, 128, 256, and 512 tokens with the number of attention units equal to the sentence length, while attention heads ranged from 1 to 8. Except for the input length experiments, the number of attention units (num_hosa) was fixed at 64. Additionally, two learning rates, 3e-4 and 3e-5, were explored to evaluate their impact on model convergence and performance.

4.3.1 Evaluation Metrics

To assess the model's performance in the next token prediction task, two key metrics were used. **Accuracy** measures the overall correctness of token predictions, while **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** evaluate n-gram overlap between the predicted and target sequences (Lin, 2004).

Among these metrics, **ROUGE-2** was selected as the primary evaluation criterion due to its sensitivity to sequence-level dependencies and its ability to robustly assess contextual relationships.

4.3.2 Experiment 1 Results and Analysis

The experimental results, summarized in Table 1, show that GPT-2-HOSA consistently outperformed the baseline GPT-2 model across all configurations. Key findings include:

GPT-2-HOSA with a learning rate of 3e-4 achieved optimal performance in smaller embedding dimensions (128 to 768). For instance, in the single-layer configuration, the ROUGE-2 score improved from 43.89 (baseline) to 77.09 (a 75.6% improvement). Conversely, a learning rate of 3e-

5 proved more effective for larger configurations (1600 to 3200), likely due to its ability to stabilize gradients and enhance convergence. Notably, GPT-2-HOSA-lr5 exhibited strong scalability, maintaining high performance even at 3200 dimensions.

GPT-2-HOSA reached peak performance between 8 and 16 layers, with scores plateauing beyond 16 layers, suggesting diminishing returns from deeper architectures. Interestingly, GPT-2-HOSA-lr5 also achieved its maximum performance at 8 layers, reinforcing the efficiency of high-order attention in moderately deep models.

Since the average sentence length in the training dataset is only 15.7 tokens, increasing the maximum sentence length beyond this average yielded limited benefits. However, as the number of self-attention heads scales with sentence length in the model design, experiments with longer sequences were conducted. Results showed that GPT-2-HOSA-lr5 continued to improve with longer sentence lengths, demonstrating the scalability and effectiveness of high-order attention under a 3e-5 learning rate.

The experimental findings validate the superior performance of HOSA, particularly in enhancing the model's ability to capture contextual relationships and scale effectively. Compared to traditional self-attention, HOSA not only excels in small parameter settings but also maintains robust improvements in high-dimensional configurations. The substantial improvements in ROUGE-2 scores further confirm HOSA's ability to model sequence-level dependencies, establishing it as a powerful enhancement over standard self-attention mechanisms.

4.4 Experiment 2: Validation of HOSA in masked token prediction

The second experiment aims to evaluate the effectiveness of HOSA in the masked token prediction pre-training task. By integrating HOSA into the scaled dot product self-attention mechanism of RoBERTa with 15% masking rate, we assessed its performance across various configurations, including dimensions $[128, 256, 768]$, layer numbers $[1, 2, 4]$, maximum sentence lengths $[64, 128, 256]$, and the number of attention heads $[1, 2, 4]$. The number of HOSA units (num_hosa) was still fixed at 64.

In the masked token prediction task, due to gradient vanishing issues observed with a learning rate of 3e-4, we adjusted the learning rate to 3e-5 to examine the impact of learning rate on model convergence and performance.

The performance of the models was evaluated using accuracy scores, which reflect the model's ability to predict masked tokens effectively.

4.4.1 Experiment 2 Results and Analysis

The experimental results, summarized in Table 2, demonstrate the advantages of RoBERTa-HOSA across various configurations. Notably, RoBERTa-HOSA shows significant improvements in small-scale models, with the performance gap narrowing as the number of layers increases.

In single-layer settings, RoBERTa-HOSA achieves substantial accuracy gains, with improvements of 24.2 and 29 points compared to the baseline—a relative improvement of 35%. These results highlight the effectiveness of HOSA in scenarios where model capacity is limited. Similarly, in multi-head configurations, although the improvements are less pronounced, RoBERTa-HOSA still outperforms the baseline (29.1 vs. 27.4). Under low-dimensional settings, the model achieves notable gains (18.5 vs. 14.8), likely due to the increased number of high-order attention heads, which enables the capture of diverse and complex attention patterns.

Consistent with the findings in Experiment 1, longer sentence lengths lead to enhanced performance. While the baseline RoBERTa experiences a performance drop at a sentence length of 256, RoBERTa-HOSA continues to improve. This can be attributed to the larger number of high-order attentions in longer sequences, which enable the model to better capture complex dependencies and patterns.

Overall, RoBERTa-HOSA demonstrates exceptional performance in small-scale models, confirming its efficiency in representation learning. While improvements in larger models are less pronounced compared to Experiment 1, the significant performance gains validate the effectiveness of the HOSA mechanism, not only in GPT pre-training tasks but also in RoBERTa-based pre-training settings.

5 Ablation Study

To better understand the effects of different model structures, we separately removed the decoder and edge position embedding in masked token prediction training to observe the effectiveness of the model.

Model	**Layer1**	**Layer2**	**Layer4**	**Layer8**	**Layer16**
GPT-2-lr4	43.89	50.69	32.66	28.75	26.72
GPT-2-lr5	34.75	43.73	60.41	72.71	74.26
GPT-2-HOSA-lr4	**77.09**	**84.93**	**93.71**	**94.56**	**94.55**
GPT-2-HOSA-lr5	42.12	58.73	90.99	94.54	**94.55**
Model	**128 Dim**	**256 Dim**	**768 Dim**	**1600 Dim**	**3200 Dim**
GPT-2-lr4	26.91	33.12	43.89	23.33	23.47
GPT-2-lr5	24.04	25.74	34.75	59.96	23.17
GPT-2-HOSA-lr4	**32.41**	**46.41**	**77.09**	22.69	23.18
GPT-2-HOSA-lr5	29.23	32.32	42.12	**78.84**	**83.35**
Model	**64 Length**	**128 Length**	**256 Length**	**512 Length**	**-**
GPT-2-lr4	43.89	27.44	25.34	23.35	-
GPT-2-lr5	34.75	34.7	34.73	40.07	-
GPT-2-HOSA-lr4	**77.09**	**52.27**	44.99	23.9	-
GPT-2-HOSA-lr5	42.12	43.58	**54.96**	**64.61**	-
Model	**1 Head**	**2 Heads**	**4 Heads**	**8 Heads**	**-**
GPT-2-lr4	43.89	52.13	56.17	61.72	-
GPT-2-lr5	34.75	35.08	36.27	38.03	-
GPT-2-HOSA-lr4	**77.09**	**77.02**	**78.59**	**85.01**	-
GPT-2-HOSA-lr5	42.12	47.7	49.36	58.24	-

Table 1: Result of the Next Token Prediction Task in Experiment 1 with ROUGE-2 scores. In experiments with varying input lengths, the number of attention units computations scales proportionally with the sentence length. Except for the input length experiments, the number of attention units is fixed at 64. The ratio between performance scores and model parameters is provided in Appendix A.

model	**layer1**	**layer2**	**layer4**
RoBERTa	24.21	27.85	**34.17**
RoBERTa-HOSA	**28.98**	**32.64**	33.98
model	**128 Dim**	**256 Dim**	**768 Dim**
RoBERTa	14.75	17.02	24.21
RoBERTa-HOSA	**18.54**	**18.05**	**28.98**
model	**64 Len**	**128 Len**	**256 Len**
RoBERTa	24.21	27.85	23.76
RoBERTa-HOSA	**28.98**	**28.76**	**32.13**
model	**1 Head**	**2 Heads**	**4 Heads**
RoBERTa	24.21	26.74	27.37
RoBERTa-HOSA	**28.98**	**28.11**	**29.11**

Table 2: Result of Masked Token Prediction Task in experiment 2 with accuracy score. This experiment is using learning rate with 3e-5 and number of attention units with 64.

Model	$3e^{-4}$	$3e^{-5}$
GPT-2	43.89	34.75
GPT-2-HOSA	77.09	42.12
GPT-2-HOSA-NS2	58.73	36.24
GPT-2-HOSA-NS4	59.11	36.37
GPT-2-HOSA-NoToken	**78.11**	**42**

Table 3: Ablation study result for Next Token Prediction task with ROUGE-2 score. NS 2 and NS 4 represent attention number ability in ablation 1, NoToken represent HOPE without token ability in ablation 2.

5.1 Ablation 1: HOPE with Restricted Ability of Self-Attention Units

To determine if accuracy improvements stem from the variation introduced by multiple self-attentions, we reduced the number of self-attentions units to 2 and 4 and observed their impact. The experiments were conducted using GPT-2-HOSA. The results are shown in Table 3.

The results confirm that training ROUGE-2 score increases as the number of self-attentions units grows, validating the importance of the High-Order effect in driving performance improvements.

5.2 Ablation 2: HOPE Without Token Distinction Ability

We explored whether unique embeddings for each token are necessary, or if shared embeddings per self-attention layer suffice. In GPT-2-HOSA, we tested shared embeddings replicated across tokens, adjusting HOPE dimensions to $[\text{batch\_size}, \text{num\_hosa}, 1, \text{hidden\_dim}]$ and replicating max_len times along the third dimension. Results are also summarized in Table 3.

The findings suggest that shared embeddings improve differentiation across self-attention layers, potentially because large models already encode token positional information effectively. Unique embeddings may add unnecessary complexity and computational cost without clear benefits. Further studies are needed to assess whether token distinction is beneficial in higher-parameter models.

6 Discussion

6.1 Why Does HOSA Significantly Enhance Performance?

We consider HOSA to greatly improve the model's performance primarily due to its ability to generate multiple self-attention mechanisms and utilize HOPE to differentiate individual attentions, tokens within those attentions, and tensors within those tokens. This approach enables the simultaneous training of multiple self-attentions while assigning distinct weights to each self-attention. The optimal attention representation is achieved by summarizing these features through a fully connected layer. Additionally, the increased data differentiation allows the model to generate larger losses during the initial training phase, which helps to adjust the weight parameters more effectively and enhances the model's learning efficiency.

6.2 Is HOSA Essentially the Same as Multi-Head?

We posit that while HOSA and the multi-head mechanism share some conceptual similarities, their implementations and computational implications are fundamentally different. While the multi-head mechanism prevents attention collapse (excessive concentration of a token's attention on itself at the expense of other relevant tokens), HOSA achieves diversified attention by replicating and assigning different weights to (k, q). The differences are evident in the computational parameters and performance. For example, experimental results show that the differences between GPT-2-lr4 and GPT-2-HOSA-lr4 with 2 and 4 heads are 7.74% (52.13 vs. 56.17) and 2.04% (78.59 vs. 77.02), respectively. Meanwhile, the difference between GPT-2-HOSA-NS2 and GPT-2-HOSA-NS4 is only 0.65% (58.73 vs. 59.11). These findings confirm that the mechanisms are fundamentally distinct.

6.3 Why Does the Learning Rate Significantly Affect Results?

We believe the core reason lies in the interaction between learning rate, hidden dimension size, and gradient stability. At high learning rates, GPT-2 models with lower hidden dimensions tend not to suffer from gradient explosion, maintaining stable training performance. However, as the hidden dimension increases, the risk of gradient explosion also increases, leading to degraded results. In contrast, HOSA demonstrates greater robustness to gradient instability at higher dimensions, which allows it to benefit more from larger learning rates. As a result, HOSA achieves better performance under higher learning rates compared to the standard GPT-2.

7 Conclusion and Future Work

HOSA is an innovative attention mechanism that integrates multiple self-attention layers into a unified structure, enabling efficient aggregation and summarization of information. This design significantly improves the accuracy and learning capabilities of self-attention-based language models, including GPT-2 and RoBERTa. Our ablation studies validate that each component of HOSA contributes effectively to the model's performance. However, the interaction between HOPE and token position embeddings may introduce conflicts under certain conditions, which warrants further exploration. Additionally, HOSA introduces a new training parameter, adding a novel dimension of depth to the training of self-attention-based models.

For future work, we plan to explore the application of HOSA in more advanced architectures, to further validate its scalability and generalization. Additionally, we will employ heatmap-based visualization to observe and track the evolution of attention weights and gradient flows throughout training, enabling a deeper understanding of parameter dynamics and facilitating more targeted optimization. Notably, HOPE has demonstrated better-than-expected performance even when combined with single-token embeddings, a phenomenon that merits further investigation. Moreover, we are interested in exploring the feasibility of extending HOSA to a four-dimensional structure, which could unlock new opportunities for enhancing both its modeling capacity and flexibility.

8 Limitation

Training Efficiency Benchmark

Another limitation is that this study focuses on evaluating the proposed attention mechanism in the pre-training phase using a limited-scale dataset and

Model	Batch Size	Throughput
GPT-2 (Baseline)	400	35,200
GPT-HOSA-NS2	380	34,304
GPT-2-HOSA	200	22,528

Table 4: Training Efficiency Comparison on A40-48GB GPU (100,000 samples)

Scalability: Each additional `attention number` unit reduces inference speed by 896 tokens/sec to 201 tokens/sec (2.5% to 1%) in a single GPU. The standard GPT-2-HOSA configuration uses 64 `attention number` units.

without downstream fine-tuning tasks. This design choice was made to isolate and analyze the fundamental behavior of the mechanism in a controlled, resource-efficient setting. Comprehensive benchmarking against established datasets (e.g., GLUE or SQuAD) typically requires extensive computational resources and prolonged experimentation cycles, which are beyond the scope of this initial investigation. While the current setup does not assess the absolute language understanding capability of the model, the observed improvements in core language modeling tasks demonstrate the mechanism's potential. We consider this work as an early-stage exploration, laying the foundation for future studies involving larger-scale training and fine-tuning.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Binghui Chen, Weihong Deng, and Jiani Hu. 2019. Mixed high-order attention network for person reidentification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 371–381.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Ashish Vaswani and 1 others. 2021. Scaling local self-attention for parameter efficient visual backbones. *CVPR*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Dan Zhang, Junfeng Shao, Xia Li, Yaqian Liu, and Liangpei Zhang. 2020. Remote sensing image super-resolution via mixed high-order attention network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(6):5183–5196.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Appendix A: Ratio between ROUGE-2 and model parameters.

	Layer = 1		
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	77.09	90.67M	8.50E-07
GPT-2-HOSA-lr5	42.12	90.67M	4.65E-07
GPT-2-lr4	43.89	84.38M	5.20E-07
GPT-2-lr5	34.75	84.38M	4.12E-07
	Layer = 2		
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	84.93	97.76M	**8.69E-07**
GPT-2-HOSA-lr5	58.73	97.76M	6.01E-07
GPT-2-lr4	50.69	91.47M	5.54E-07
GPT-2-lr5	43.73	91.47M	4.78E-07
	Layer = 4		
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	93.71	111.94M	8.37E-07
GPT-2-HOSA-lr5	90.99	111.94M	8.13E-07
GPT-2-lr4	32.66	105.65M	3.09E-07
GPT-2-lr5	60.41	105.65M	5.72E-07
	Layer = 8		
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	94.56	140.29M	6.74E-07
GPT-2-HOSA-lr5	94.54	140.29M	6.74E-07
GPT-2-lr4	28.75	134.00M	2.15E-07
GPT-2-lr5	72.71	134.00M	5.43E-07
	Layer = 16		
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	94.55	196.99M	4.80E-07
GPT-2-HOSA-lr5	94.55	196.99M	4.80E-07
GPT-2-lr4	26.72	190.70M	1.40E-07
GPT-2-lr5	74.26	190.70M	3.89E-07

Table 5: Model performance comparison across different Transformer layer depths. Rows are grouped by the number of layers for clearer comparison.

Model	Score	Params	Score/Param
Heads = 1			
GPT-2-HOSA-lr4	77.09	90.67M	8.50E-07
GPT-2-HOSA-lr5	42.12	90.67M	4.65E-07
GPT-2-lr4	43.89	84.38M	5.20E-07
GPT-2-lr5	34.75	84.38M	4.12E-07
Heads = 2			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	77.02	90.67M	8.49E-07
GPT-2-HOSA-lr5	47.70	90.67M	5.26E-07
GPT-2-lr4	52.13	84.38M	6.18E-07
GPT-2-lr5	35.08	84.38M	4.16E-07
Heads = 4			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	78.59	90.67M	8.67E-07
GPT-2-HOSA-lr5	49.36	90.67M	5.44E-07
GPT-2-lr4	56.17	84.38M	6.66E-07
GPT-2-lr5	36.27	84.38M	4.30E-07
Heads = 8			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	85.01	90.67M	**9.38E-07**
GPT-2-HOSA-lr5	58.24	90.67M	6.42E-07
GPT-2-lr4	61.72	84.38M	7.31E-07
GPT-2-lr5	38.03	84.38M	4.51E-07

Table 6: Model score, parameter count, and score-to-parameter ratio under different numbers of attention heads. Rows are grouped by head count.

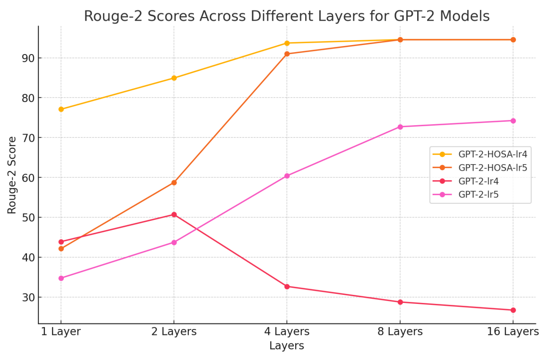

Figure 4: GPT-2 Model ROUGE-2 Scores Across Different Layers

B Appendix B: Complete score result.

C Appendix C: Graphic score result.

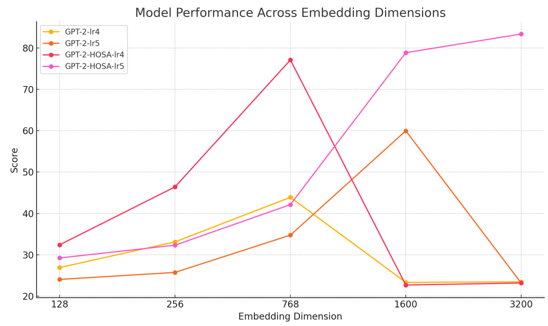

Figure 5: GPT-2 Model ROUGE-2 Across Embedding Dimensions

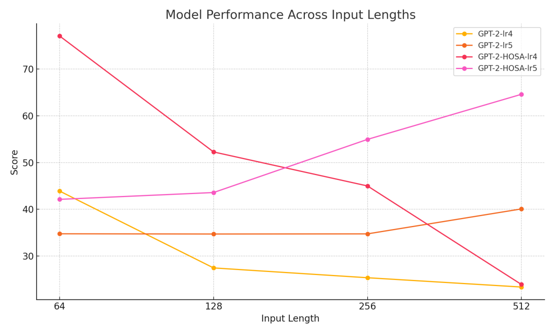

Figure 6: GPT-2 Model ROUGE-2 Across Input Lengths

Figure 7: GPT-2 Model ROUGE-2 Across Number Of Attention Heads

Model	Score	Params	Score/Param
Dimension = 128			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	32.41	13.15M	**2.47E-06**
GPT-2-HOSA-lr5	29.23	13.15M	2.22E-06
GPT-2-lr4	26.91	13.08M	2.06E-06
GPT-2-lr5	24.04	13.08M	1.84E-06
Dimension = 256			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	46.41	26.55M	1.75E-06
GPT-2-HOSA-lr5	32.32	26.55M	1.22E-06
GPT-2-lr4	33.12	26.55M	1.25E-06
GPT-2-lr5	25.74	26.55M	9.69E-07
Dimension = 768			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	77.09	90.67M	8.50E-07
GPT-2-HOSA-lr5	42.12	90.67M	4.65E-07
GPT-2-lr4	43.89	84.38M	5.20E-07
GPT-2-lr5	34.75	84.38M	4.12E-07
Dimension = 1600			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	22.69	192.59M	1.18E-07
GPT-2-HOSA-lr5	78.84	192.59M	4.09E-07
GPT-2-lr4	23.33	191.77M	1.22E-07
GPT-2-lr5	59.96	191.77M	3.13E-07
Dimension = 3200			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA-lr4	23.18	446.62M	5.19E-08
GPT-2-HOSA-lr5	83.35	446.62M	1.87E-07
GPT-2-lr4	23.47	444.98M	5.27E-08
GPT-2-lr5	23.17	444.98M	5.21E-08

Table 7: Model performance comparison under different hidden dimensions. Rows are grouped by dimension.

Model	Score	Params	Score/Param
Learning Rate = 3e-4			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA	77.09	90.67M	8.50E-07
GPT-2-HOSA-NS2	58.73	84.58M	6.94E-07
GPT-2-HOSA-NS4	59.11	84.78M	6.97E-07
GPT-2-HOSA-Dim1	65.61	–	–
GPT-2-HOSA-NoToken	78.11	84.43M	**9.25E-07**
GPT-2	43.89	84.38M	5.20E-07
Learning Rate = 3e-5			
Model	**Score**	**Params**	**Score/Param**
GPT-2-HOSA	42.12	90.67M	4.65E-07
GPT-2-HOSA-NS2	36.24	84.58M	4.28E-07
GPT-2-HOSA-NS4	36.37	84.78M	4.29E-07
GPT-2-HOSA-Dim1	40.00	–	–
GPT-2-HOSA-NoToken	42.00	84.43M	4.97E-07
GPT-2	34.75	84.38M	4.12E-07

Table 8: Model performance comparison under different learning rates. Rows are grouped by learning rate setting.

Table 9: Layer-wise Evaluation Results (GPT-2-HOSA-lr4/lr5 vs GPT-2-lr4/lr5)

		Layer = 1		
Model	**Acc**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
GPT-2-HOSA-lr4	77.09	84.58	77.09	84.46
GPT-2-HOSA-lr5	42.12	62.81	42.12	61.57
GPT-2-lr4	43.89	65.31	43.89	63.54
GPT-2-lr5	34.75	57.61	34.75	55.80
		Layer = 2		
GPT-2-HOSA-lr4	84.93	88.39	84.93	88.37
GPT-2-HOSA-lr5	58.73	73.91	58.73	73.31
GPT-2-lr4	50.69	69.71	50.69	68.49
GPT-2-lr5	43.73	64.18	43.73	64.14
		Layer = 4		
GPT-2-HOSA-lr4	93.71	93.83	93.71	93.83
GPT-2-HOSA-lr5	90.99	92.45	90.99	92.45
GPT-2-lr4	32.66	57.57	32.66	54.90
GPT-2-lr5	60.41	73.86	60.41	73.56
		Layer = 8		
GPT-2-HOSA-lr4	94.56	94.51	94.56	94.51
GPT-2-HOSA-lr5	94.54	94.49	94.54	94.49
GPT-2-lr4	28.75	54.01	28.75	51.07
GPT-2-lr5	72.71	80.67	72.71	80.57
		Layer = 16		
GPT-2-HOSA-lr4	94.55	94.50	94.55	94.50
GPT-2-HOSA-lr5	94.55	94.50	94.55	94.50
GPT-2-lr4	26.72	51.84	26.72	48.89
GPT-2-lr5	74.26	81.54	74.26	81.45

Table 10: Dimension-wise Evaluation Results (GPT-2-HOSA-lr4/lr5 vs GPT-2-lr4/lr5)

		Dim = 128		
Model	**Acc**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
GPT-2-HOSA-lr4	32.41	53.39	32.41	51.41
GPT-2-HOSA-lr5	29.23	51.35	29.23	48.96
GPT-2-lr4	26.91	51.19	26.91	48.30
GPT-2-lr5	24.04	47.68	24.04	44.62
		Dim = 256		
GPT-2-HOSA-lr4	46.41	65.09	46.41	64.12
GPT-2-HOSA-lr5	32.32	53.37	32.32	51.39
GPT-2-lr4	33.12	56.81	33.12	54.42
GPT-2-lr5	25.74	49.75	25.74	46.84
		Dim = 1600		
GPT-2-HOSA-lr4	22.69	48.58	22.69	45.84
GPT-2-HOSA-lr5	78.84	84.93	78.84	84.88
GPT-2-lr4	23.33	48.99	23.33	46.55
GPT-2-lr5	59.96	73.49	59.96	73.07
		Dim = 3200		
GPT-2-HOSA-lr4	23.18	49.04	23.18	46.67
GPT-2-HOSA-lr5	83.35	87.38	83.35	87.35
GPT-2-lr4	23.47	49.15	23.47	46.88
GPT-2-lr5	23.17	48.70	23.17	46.18

Table 11: Length-wise Evaluation Results (GPT-2-HOSA-lr4/lr5 vs GPT-2-lr4/lr5). The number of attention units is proportional to the sentence length.

Model	Acc	ROUGE-1	ROUGE-2	ROUGE-L
		Length = 128		
GPT-2-HOSA-lr4	95.47	71.77	52.27	70.43
GPT-2-HOSA-lr5	94.25	63.07	43.58	62.15
GPT-2-lr4	91.65	52.39	27.44	49.52
GPT-2-lr5	91.37	51	26.83	48.46
		Length = 256		
GPT-2-HOSA-lr4	88.26	64.13	44.99	63.04
GPT-2-HOSA-lr5	97.88	70.52	54.96	70.19
GPT-2-lr4	97.45	67.62	47.39	66.07
GPT-2-lr5	96.51	57.61	34.73	55.81
		Length = 512		
GPT-2-HOSA-lr4	97.68	48.30	23.90	45.72
GPT-2-HOSA-lr5	99.20	76.57	64.61	76.36
GPT-2-lr4	97.71	48.77	23.35	46.27
GPT-2-lr5	98.47	61.56	40.07	60.10

Table 12: Head-wise Evaluation Results (GPT-2-HOSA-lr4/lr5 vs GPT-2-lr4/lr5)

Model	Acc	ROUGE-1	ROUGE-2	ROUGE-L
		Heads = 2		
GPT-2-HOSA-lr4	77.02	84.43	77.02	84.65
GPT-2-HOSA-lr5	47.70	66.61	47.70	65.37
GPT-2-lr4	52.13	70.08	52.13	68.92
GPT-2-lr5	35.08	57.80	35.08	56.05
		Heads = 4		
GPT-2-HOSA-lr4	78.59	85.85	78.59	85.68
GPT-2-HOSA-lr5	49.36	67.63	49.36	66.80
GPT-2-lr4	56.17	72.18	56.17	71.29
GPT-2-lr5	36.27	58.83	36.27	57.19
		Heads = 8		
GPT-2-HOSA-lr4	85.01	88.99	85.01	88.94
GPT-2-HOSA-lr5	58.24	74.42	58.24	73.95
GPT-2-lr4	61.72	57.19	61.72	74.57
GPT-2-lr5	38.03	59.98	38.03	58.52