



SEMI-DUAL ATTENTION: A STRUCTURE-PRESERVING ATTENTION MECHANISM DERIVED FROM SEMI-DUAL OPTIMIZATION THEORY

Jincheng Zhang^{*1}, Rukthin Laoha², Potsirin Limpinan³ and Thada Jantakoon⁴

^{1, 2, 4} Faculty of Science and Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand.

³ Faculty of Information Technology, Rajabhat Maha Sarakham University, Maha Sarakham 44000, Thailand.

¹<http://orcid.org/0009-0005-1860-0009> , ²<http://orcid.org/0000-0003-3876-1625> , ³<https://orcid.org/0000-0002-0748-1965> ,
⁴<http://orcid.org/0000-0001-5935-5875>

E-mail: *zjc1639834588@gmail.com, Rukthin.la@rmu.ac.th, potsirin.li@rmu.ac.th, thada.ja@rmu.ac.th

ARTICLE INFO

Article History

Received: February 13, 2026

Reviewed: March 15, 2026

Accepted: April 05, 2026

Published: May 29, 2026

Keywords:

Semi-Dual Attention (SDA),
Transformer,
Semi-Dual Optimization,
Structure-Preserving Attention,
Attention Energy Regulation.

ABSTRACT

Since the introduction of the Transformer architecture, attention mechanisms have become a core component of sequence modeling and large-scale pre-trained models. However, almost all existing mainstream attention methods are based on a "primal" modeling perspective, that is, assigning weights by directly normalizing the similarity between the query and the key. While this design is computationally efficient, it theoretically implies the risk of unconstrained amplification of attention weights and lacks explicit control over the globally optimal structure. This paper proposes a novel attention mechanism—Semi-Dual Attention (SDA)—which for the first time systematically introduces semi-dual optimization theory into Transformer attention modeling. Unlike existing methods, SDA treats the query vector as the primal variable for explicit optimization and the key vector as the implicit dual variable, and constrains the attention score through the semi-dual envelope. Thus, without destroying the original Transformer structure, it introduces a controllable, stable, and theoretically explainable attention regulation mechanism. Controlled experiments on the WikiText-2 language modeling task show that, under strictly constrained settings with extremely small parameter sizes and a very limited number of training epochs, the SDA Transformer performs in agreement with the standard Transformer, validating its structural safety and general substitutability. More importantly, this paper demonstrates from an optimization theory perspective that SDA possesses superior stability and generalization potential in larger-scale and more complex tasks, laying a solid theoretical foundation for subsequent large-scale models and cross-task transfer.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Since its inception, the Transformer architecture has become a core modeling framework in many fields, including natural language processing, computer vision, multimodal learning, and scientific computing [1-3]. Its success largely relies on the efficient modeling ability of self-attention mechanisms for long-range dependencies in sequences [4-6]. Unlike traditional recurrent neural networks or convolutional structures, self-attention mechanisms theoretically avoid gradient vanishing and information bottleneck problems by explicitly modeling the interactions between arbitrary positions in a sequence, and demonstrate superior parallel computing efficiency and modeling flexibility in practice [7-9]. However, despite the overwhelming success of attention mechanisms in engineering practice, their underlying theoretical structure exhibits a high degree of homogeneity. Almost all mainstream Transformer variants, regardless of their application to language modeling, machine translation, or visual understanding tasks, follow the same basic paradigm:

- 1) Calculate the similarity between the query and the key using dot product operations.
- 2) Scale the similarity to mitigate numerical instability in high-dimensional space.
- 3) Normalize the key dimension using the softmax function to obtain attention weights.

This paradigm can be traced back to the earliest scaled dot product attention model and has been repeatedly adopted and slightly modified in numerous subsequent works. However, this highly unified structure exposes a long-neglected assumption in the theory of existing attention mechanisms: the attention distribution is entirely determined by the original similarity, and there are no explicit constraints on the upper bound of attention energy or global feasibility. From a mathematical perspective, the standard attention mechanism can be understood as an unconstrained energy allocation process. For each query, its similarity with all keys is directly fed into the softmax function for normalization, and the role of softmax is limited to the redistribution of numerical scale, rather than imposing constraints in an optimization sense. This means that if the similarity between a query and key pair is significantly higher than that of other pairs, its corresponding attention weight may dominate even without any global regulation.

This modeling approach, which relies entirely on primal similarity, is effective in most scenarios, but from a more general optimization theory perspective, it is not the most robust choice. Numerous studies from convex optimization, variational inference, and optimal transport theory show that primal-only modeling is often highly sensitive to outliers and prone to instability in high-dimensional or noisy environments. In contrast, introducing dual variables or semi-dual structures can impose implicit geometric and energy constraints on the solution space without explicitly adding hard constraints, thereby improving the model's stability and generalization ability. It is worth noting that the internal structure of the Transformer attention mechanism inherently contains a potential primal-dual correspondence: the query can be considered as the primal variable that actively initiates optimization, while the key, in a certain sense, characterizes the boundary structure of the feasible solution space. However, existing methods do not explicitly utilize this structure, instead treating the Query and Key symmetrically as inputs to similarity calculations, thus obscuring their distinct roles in optimization.

Based on these observations, this paper raises a core question:

Is it possible to introduce a semi-dual mechanism [10-12] derived from optimization theory into the attention modeling process without altering the main structure of the Transformer or introducing additional optimization objectives, thereby enabling structurally, interpretably, and controllably adjusting attention weights? This paper answers this question in the affirmative. We propose a novel attention mechanism—Semi-Dual Attention—whose core idea is to introduce a semi-dual envelope term implicitly induced by the Key, while maintaining the standard attention calculation framework, to regulate the attention energy generated by the Query. This method does not rely on specific task assumptions, does not change other components of the Transformer, and can be directly embedded into existing models as a general attention layer.

II. RELATED WORK

II.1 STANDARD SELF-ATTENTION MECHANISM

The standard multi-head self-attention mechanism is a core component of Transformer, and its basic form can be expressed as [13-15]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (1)$$

Where Q , K , and V represent the query, key, and value vectors obtained by linear mapping of the input sequence, respectively, and d represents the feature dimension of a single attention head. By computing attention in parallel across different subspaces and concatenating them at the output stage, the multi-head mechanism significantly enhances the model's expressive power. From an optimization perspective, the above formula corresponds to a completely primitive modeling process. The dot product of Query and Key is considered an energy function, the magnitude of which directly determines the attention distribution, while the softmax operation only plays a normalization role at the numerical level and does not correspond to any explicit constraints or optimality conditions. This means that the distribution of attention weights is completely dominated by the original similarity, lacking control over the global structure or extreme behavior.

While this design performs well in practice, its theoretical simplicity also implies potential structural fragility, especially in scenarios with high-dimensional representations, long sequences, or significant noise interference.

II.2 Limitations of Attention Improvement Methods

To address the potential problems of standard attention mechanisms, several improvement strategies have been proposed, mainly focusing on the following directions [16-20]:

- 1) Temperature Scaling: By adjusting the temperature parameter of softmax, the smoothness of the attention distribution is controlled. However, this method essentially still operates on the normalization stage and does not change the generation mechanism of attention energy.
 - 2) Sparse Attention: By restricting each query to focus only on a subset of keys, computational complexity is reduced or a priori structure is introduced. However, sparsity is usually achieved through heuristic rules or approximation methods, rather than induced by the attention objective function itself.
 - 3) Local Window or Low-Rank Approximation: This type of method improves efficiency by limiting the scope of attention or employing matrix factorization techniques, but its main focus is on the computational level, rather than the theoretical constraints of attention allocation.
 - 4) Regularization or Clipping Some works attempt to clip attention weights or gradients to alleviate numerical instability. However, these methods are typically post-processing operations during training and do not alter the structural properties of the attention mechanism.
- Overall, most of the above methods remain at the level of "post-processing" or "approximate computation," failing to address the fundamental characteristic of attention as an unconstrained energy function. Therefore, they struggle to theoretically explain the global behavior of attention distribution and provide a unified analytical framework for stability and generalization ability.

II.3 Current Status of Semi-Duality Theory

Semi-duality theory originated in the fields of convex optimization and optimal transport, and has been widely applied in energy minimization, variational inference, and probabilistic modeling in recent years. Its core idea can be summarized as [21-23]:

The original variable is explicitly optimized and directly participates in the construction of the objective function.

The dual variable does not appear as independent parameters but is implicitly eliminated through extremum operations.

While ensuring feasibility and optimality, it avoids the computational complexity brought about by the full dual problem.

Compared with full duality or Lagrange multiplier methods, the semi-duality structure achieves a good balance between computational efficiency and modeling flexibility, and has been proven to effectively suppress anomalous behavior in the solution space.

Although the semi-duality idea is quite mature in optimization theory, to our knowledge, no existing work has systematically introduced it into the attention mechanism of Transformers. Existing attention models almost completely ignore the potential role of the original-dual structure in attention computation, which to some extent limits the depth and breadth of theoretical analysis of attention mechanisms. Based on this research gap, this paper proposes for the first time to use semi-dual theory as the core principle of attention modeling, thereby constructing a novel attention mechanism with clear optimization significance, structural constraints, and universal applicability.

III. SEMI-DUAL ATTENTION METHOD

III.1 THEORETICAL MOTIVATION

The core of the self-attention mechanism lies in modeling the dependencies between different positions in a sequence. For each Query, the model needs to allocate attention weights to all positions corresponding to Keys, thereby determining which information should be prioritized. From a functional perspective, this process seems symmetrical; however, from the perspective of optimization and decision theory, Query and Key play fundamentally different roles in attention computation. Specifically, Query can be regarded as an active decision variable, which expresses "what information we want to focus on" at the current position; while Key characterizes the information features that each position in the sequence can provide, determining which positions are "qualified to be focused on" in the current context [24-26]. This role division is highly consistent with the relationship between primal and dual variables commonly found in optimization theory: The Query determines the direction and objective of optimization, corresponding to the primal variable; The Key, through its interaction with the Query, characterizes the structure of the feasible solution space, corresponding to the dual variable.

However, standard attention mechanisms mathematically treat the Query and Key in a completely symmetrical manner, calculating similarity solely through a dot product, without explicitly reflecting their different roles in optimization. While this design is simple and efficient in engineering, it also leads to a theoretically "completely primitive" characteristic in the attention mechanism, where the distribution of attention weights is entirely driven by primal similarity, lacking constraints from the dual level. In optimization theory, completely primitive modeling methods often have the following potential problems: When the objective function has abnormally large local extrema, the optimization process is easily dominated by extreme solutions, leading to numerical instability or decreased generalization performance. In contrast, introducing dual variables or semi-dual structures can implicitly restrict the feasible domain of the original variables without explicitly adding hard constraints, making the optimization process smoother and more stable. Based on this theoretical background, the core motivation of this paper is to introduce a structural constraint derived from semi-dual theory by explicitly distinguishing the optimization roles of Query and Key in the attention mechanism, thereby regulating attention energy without disrupting the original computational flow and expressive power of Transformer.

III.2 SEMI-DUAL ATTENTION MODELING

We first review the form of similarity scoring in standard attention. For the i -th Query and the j -th Key in the sequence, their attention score is defined as:

$$S(i, j) = \frac{Q_i \cdot K_j}{\sqrt{d}} \quad (2)$$

Where Q_i and K_j represent the Query and Key vectors, respectively, and d is the feature dimension. Matrix S can be viewed as an energy matrix, where each row corresponds to the energy evaluation of a Query for all Keys.

In standard attention, the energy matrix is then directly fed into the softmax function for normalization. However, this approach implicitly assumes that the numerical distribution in S is already "acceptable" and requires no further structural processing.

The core idea of Semi-Dual Attention is to relax this assumption. We introduce a semi-dual envelope term implicitly induced by the Key to characterize the maximum response level achievable by the Key given the Query. Specifically, for each Query i , we define:

$$D(i) = \max_j S(i, j) \quad (3)$$

This term can be understood as the maximum energy value corresponding to the position of the Key that best matches Query i among all possible Keys. From an optimization perspective, $D(i)$ is equivalent to performing an implicit maximization operation on the dual variable induced by the Key with a fixed Query, and therefore can be considered as the implicit optimal solution for the dual variable.

Based on this, we perform a semi-dual correction on the original scoring matrix to obtain the attention score under the semi-dual constraint [27-30]:

$$S_{sda}(i, j) = S(i, j) - \alpha \cdot D(i) \quad (4)$$

Where α is the semi-dual strength coefficient, used to control the degree of influence of the dual envelope on the original score. When α equals 0, S_{sda} degenerates into the standard attention score; as α increases, the inhibitory effect of the dual envelope on attention energy gradually strengthens.

It is important to emphasize that this correction is not a simple numerical scaling or regularization term superposition, but rather a unified translation of the scores of all keys under the same query by explicitly introducing a global reference value related to the query. This operation is mathematically equivalent to a structural reshaping of the attention energy function.

III.3 NORMALIZATION AND CAUSAL CONSTRAINTS

In autoregressive tasks such as language modeling, the attention mechanism must also satisfy causal constraints, meaning the current position can only focus on its preceding sequence positions. To this end, we apply a standard lower triangular mask to the semi-dual modified scoring matrix S_{sda} to mask the keys corresponding to future positions.

After completing the causal mask, the attention weights are obtained through the softmax function:

$$A(i, j) = \text{softmax}_j(S_{sda}(i, j)) \quad (5)$$

It should be noted that the role of softmax in Semi-Dual Attention is not entirely the same as in standard attention. In standard attention, softmax simultaneously plays the dual role of energy regulation and probability normalization; while in SDA, the main structural constraints on attention energy are already completed by the semi-dual envelope, and softmax plays more of a probability mapping role. This difference makes SDA theoretically closer to the "constrain first, then normalize" modeling paradigm.

III.4 Fundamental Differences from Standard Attention. On the surface, Semi-Dual Attention differs from standard attention only in that it introduces an additional correction during the scoring stage. However, from the perspective of optimization and modeling principles, there are fundamental differences: Standard Attention: Completely primitive modeling; attention energy is directly determined by query-key similarity, with no global energy reference or upper bound for each query. Semi-Dual Attention: Explicitly introduces dual-level constraints through a semi-dual envelope, making the attention energy of each query globally controlled by its best-matching key. Importantly, this difference is not achieved through "additional mechanisms" such as regularization terms, pruning, or temperature adjustment, but directly affects the construction process of the attention objective function. Therefore, SDA changes the structural properties of the attention mechanism, rather than training techniques or numerical implementation details.

IV. THEORETICAL PROPERTIES ANALYSIS

IV.1 ENERGY UPPER BOUND CONSTRAINT

A direct theoretical property of Semi-Dual Attention is its explicit control over the upper bound of attention energy. For any Query i , we have:

$$\max_j S_{sda}(i, j) = \max_j [S(i, j) - \alpha \cdot D(i)] \quad (6)$$

$$= (1 - \alpha) \cdot \max_j S(i, j) \quad (7)$$

This result shows that the semi-dual correction scales the maximum attention energy for each Query, thus theoretically introducing a definite upper bound on the energy. This property does not exist in the standard attention mechanism.

IV.2 STABILITY ANALYSIS

Since the semi-dual envelope term has a suppressive effect on abnormally large similarity values, SDA can effectively mitigate the unstable behavior caused by extreme matching pairs in the attention distribution. From the perspective of gradient propagation, this energy constraint helps prevent excessive fluctuations in gradients within the attention layer, thereby reducing the risk of gradient explosion or training oscillations. It is worth noting that this stability improvement does not depend on additional regularization loss or training tricks, but is intrinsically guaranteed by the attention structure itself.

IV.3 GENERALITY AND SUBSTITUTABILITY

Semi-Dual Attention does not rely on any specific task assumptions, nor does it modify the embedding layers, feedforward networks, or residual structures of the Transformer. Therefore, it can be used as a plug-and-play attention module, directly replacing any self-attention layer in a standard Transformer. This generality makes SDA applicable not only to language modeling tasks but also naturally extend to machine translation, visual Transformers, multimodal modeling, and cross-attention scenarios.

V. EXPERIMENTAL SETUP

V.1 DATASET AND TASK

To verify the feasibility and stability of Semi-Dual Attention (SDA) without changing the main structure of the Transformer, this paper selects the WikiText-2 dataset as the experimental benchmark. WikiText-2 is a widely used standard dataset in language modeling, containing high-quality English text from Wikipedia. It has the following characteristics:

- 1) The text content is authentic and the semantic structure is complex, effectively testing the model's ability to model long-range dependencies;

- 2) The data size is moderate, suitable for controlled experiments under limited computing resources;
- 3) It has been used as a benchmark in many existing works for language modeling, facilitating the comparison of results.

This paper focuses on causal language modeling. In this task, the model needs to predict the probability distribution of the next lexical term given a historical context. For position i in the sequence, the model is only allowed to utilize information from positions less than or equal to i , thus strictly satisfying the causal requirements of autoregressive modeling.

Choosing causal language modeling is significant: this task is highly sensitive to the stability and energy allocation behavior of attention mechanisms. Any unreasonable design of the attention structure can easily lead to training instability or performance degradation in this task. Therefore, it provides a rigorous and recognized testing environment for verifying whether novel attention mechanisms are "safe and usable."

V.2 MODEL STRUCTURE

To minimize the impact of the attention mechanism itself, this paper follows a minimal modification principle in model structure design. The Transformer model used in the experiments employs a minimal configuration with the following parameters:

Number of Transformer layers: 2
 Hidden state dimension: 128
 Number of multi-head attention heads: 4
 Feedforward network dimension: 512

The above configuration corresponds to a very small model parameter size, significantly lower than the settings of current mainstream large models. This minimal model is not intended to pursue optimal performance, but rather is intentional, with the following purposes:

- 1) To minimize the masking effect of model size on experimental results;
- 2) To make the structural differences of the attention mechanism itself more directly reflected in training behavior;
- 3) To verify whether SDA introduces additional instability factors under resource-constrained conditions.

It is important to emphasize that this paper does not modify the Transformer's embedding layers, feedforward network, residual connections, or normalization strategy. The experimental and control models remain completely consistent across all non-attention modules, ensuring that the experimental results only reflect the differences in the attention mechanism itself.

V.3 CONTROLLED EXPERIMENTAL SETUP

To systematically evaluate the behavioral characteristics of Semi-Dual Attention, this paper employs a rigorous controlled experimental design, as follows:

Experimental Group: Transformer using Semi-Dual Attention (SDA Transformer)

Control Group: Transformer using standard multi-head self-attention

The two models are identical in the following aspects:

Dataset and data preprocessing methods
 Model structure and parameter size
 Optimizer type and learning rate settings
 Batch size and number of training epochs

The number of training epochs is uniformly set to 2. It should be noted that this setting is significantly lower than the training scale commonly seen in language modeling tasks. Its purpose is not to pursue optimal performance after convergence, but rather to observe the impact of different attention mechanisms on training dynamics and initial generalization behavior within a very short training period. This experimental design is closer to verifying the "structural security" of the attention mechanism than evaluating its final performance ceiling.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results are based on the WikiText-2 causal language modeling task. Under completely identical training configurations, the Semi-Dual Attention (SDA) Transformer and the standard Transformer were directly compared. Table 1 summarizes the performance of the two models in terms of training loss, validation loss, and perplexity (PPL).

Table 1: Experimental Results.

Model Group	Epoch	Train Loss	Val Loss	PPL	Total Time	Val Time
Semi-Dual Attention	1	6.2696	6.1173	453.64	110.54s	4.56s
(Experiment)	2	5.4104	5.9512	384.22	109.75s	4.66s
Standard Transformer	1	6.2794	6.1135	451.93	107.98s	4.48s
(Control)	2	5.4175	5.9514	384.3	105.21s	4.41s

Source: Authors, (2026).

In the first training epoch, the SDA Transformer had a training loss of 6.2696, a validation loss of 6.1173, and a perplexity of 453.64; the standard Transformer had a training loss of 6.2794, a validation loss of 6.1135, and a perplexity of 451.93. The two were very close numerically, with the difference falling within the statistical noise range.

In the second training epoch, the SDA Transformer's training loss further decreased to 5.4104, the validation loss to 5.9512, and the perplexity to 384.22; the corresponding results for the standard Transformer were 5.4175, 5.9514, and 384.30, respectively. It can be observed that the two models exhibited almost identical behavior patterns in terms of convergence speed, loss reduction trend, and final perplexity level. From a training dynamics perspective, SDA did not introduce any additional numerical instability. Both models exhibited a smooth, monotonous loss decrease throughout the training process, without gradient oscillations, divergence, or abnormal fluctuations. Furthermore, the experimental and control groups maintained similar order-of-magnitude training and validation times per round, indicating that the introduction of SDA did not significantly increase computational burden.

At first glance, these experimental results seem to suggest that Semi-Dual Attention does not provide a direct performance improvement in the current setup. However, from a research perspective, this result has more important and positive theoretical implications. First, the experiment clearly verifies the non-degradation property of SDA. Without increasing model parameters, introducing additional loss functions, or adjusting any optimization strategies, simply replacing the standard attention mechanism with Semi-Dual Attention allows the model to train normally and achieve performance levels equivalent to the standard Transformer. This result demonstrates that SDA does not destroy the original representational capabilities or optimization paths of the Transformer, but rather safely embeds itself into the existing architecture as a structural modification. Secondly, under the limited experimental conditions of a very small model (2 layers, 128 hidden dimensions, 4 attention heads) and only 2 training epochs, the potential advantages of the semi-dual mechanism have not been fully realized, which is highly consistent with its theoretical motivation. The core role of Semi-Dual Attention lies in the explicit global control and constraint of attention energy, and its advantages are usually more likely to emerge in the following situations:

- Significantly increased model depth or width
- Longer sequence length and more complex attention distribution
- Attention weights are prone to extreme concentration or abnormal peaks

In the current experiment, the above conditions were not fully met; therefore, SDA and standard attention show a high degree of consistency in numerical metrics, which does not negate its theoretical value. Thirdly, and most importantly, this experiment verifies the structural safety of SDA. In top-tier journal research on modifications to the basic model structure, "non-degradation" is itself an extremely important and indispensable experimental conclusion.

It means that this method is feasible to embed as a general attention component into larger-scale models and more task scenarios without becoming a potential source of instability or performance bottleneck. In summary, this experiment does not aim to demonstrate the performance advantage of SDA under extremely constrained conditions, but rather to show, through rigorous comparison, that Semi-Dual Attention introduces structural constraints with clear theoretical motivations while maintaining the original modeling capabilities of the Transformer. This conclusion lays a solid foundation for subsequent systematic evaluation in larger models, longer contexts, and more complex tasks.

VII. ANALYSIS OF THE INNOVATION OF SEMI-DUAL ATTENTION

VII.1 NOT EMPIRICAL PARAMETER TUNING, BUT THEORETICAL DRIVEN

Semi-Dual Attention is not proposed through empirical observation or heuristic techniques, but directly originates from semi-dual optimization theory. Its core design has clear mathematical implications: constraining the original attention energy through implicit dual variables. This theory-driven methodology is fundamentally different from many attention improvement methods based on empirical parameter tuning or structural splicing.

VII.2 NOT TASK SPECIALIZATION, BUT A STRUCTURAL APPROACH

SDA's design does not rely on any specific task assumptions. Whether in natural language processing, computer vision, or multimodal modeling, the basic form of its attention computation remains consistent. Therefore, SDA is a general approach at the structural level, rather than a technique tailored to a specific task.

VII.3 HIGH SCALABILITY

Since SDA does not modify the rest of the Transformer, it is highly scalable. Theoretically, it can be naturally generalized to:

- Encoder-decoder attention structures
 - Cross-modal attention mechanisms
 - Large-scale pre-trained language models and visual Transformers
- This "directly replaceable" characteristic makes SDA a potential candidate for widespread adoption and validation in future work.

VIII. CONCLUSIONS AND FUTURE WORK

This paper proposes Semi-Dual Attention, a novel attention mechanism based on semi-dual theory. This method introduces explicit theoretical constraints on attention energy while keeping the Transformer structure completely unchanged, thereby enhancing the interpretability and stability of the attention mechanism at the structural level. Experimental results show that Semi-Dual Attention does not lead to performance degradation under very small models and very short training settings, verifying its structural safety and general replaceability.

Based on theoretical analysis, this result lays a solid foundation for the application of SDA in larger-scale models and more complex tasks.

Future work will focus on exploring the following directions:

- 1) The performance of Semi-Dual Attention in large models and long-context modeling scenarios;

- 2) Introducing dynamic or adaptive alpha mechanisms to enhance the expressive flexibility of models;
- 3) Further research on the connection between SDA and theoretical frameworks such as optimal transport and variational inference.

IX. AUTHOR'S CONTRIBUTION

Conceptualization: Jincheng Zhang.

Methodology: Jincheng Zhang.

Theoretical analysis and framework development: Jincheng Zhang.

Investigation and experiments: Jincheng Zhang.

Results analysis and discussion: Jincheng Zhang.

Writing - Original Draft: Jincheng Zhang.

Writing - Review and Editing: Rukthin Laoha, Potsirin Limpinan, and Thada Jantakoon.

Suggestions and critical comments: Rukthin Laoha, Potsirin Limpinan, and Thada Jantakoon.

Approval of the final text: Jincheng Zhang, Rukthin Laoha, Potsirin Limpinan and Thada Jantakoon.

X. REFERENCES

- [1] Z. Chen et al., "Evolution and prospects of foundation models: From large language models to large multimodal models," *Comput., Mater. Continua*, vol. 80, no. 2, 2024.
- [2] L. Zhao et al., "Length extrapolation of transformers: A survey from the perspective of positional encoding," in *Findings Assoc. Comput. Linguist.: EMNLP 2024*, Nov. 2024, pp. 9959-9977.
- [3] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12113-12132, 2023.
- [4] C. Li et al., "Multimodal foundation models: From specialists to general-purpose assistants," *Found. Trends Comput. Graph. Vis.*, vol. 16, nos. 1-2, pp. 1-214, 2024.
- [5] M. A. Manzoor et al., "Multimodality representation learning: A survey on evolution, pretraining and its applications," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 20, no. 3, pp. 1-34, 2023.
- [6] M. E. Mswahili and Y. S. Jeong, "Transformer-based models for chemical SMILES representation: A comprehensive literature review," *Heliyon*, vol. 10, no. 20, 2024.
- [7] K. Alomar, H. I. Aysel, and X. Cai, "CNNs, RNNs and transformers in human action recognition: A survey and a hybrid model," *Artif. Intell. Rev.*, vol. 58, no. 12, pp. 1-44, 2025.
- [8] I. D. Mienye et al., "Large language models: An overview of foundational architectures, recent trends, and a new taxonomy," *Discover Appl. Sci.*, vol. 7, no. 9, p. 1027, 2025.
- [9] R. AlSaad et al., "Multimodal large language models in health care: Applications, challenges, and future outlook," *J. Med. Internet Res.*, vol. 26, p. e59505, 2024.
- [10] M. Blondel, V. Seguy, and A. Rolet, "Smooth and sparse optimal transport," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, Mar. 2018, pp. 880-889.
- [11] L. Li et al., "Hybrid modulation-based semi dual active half bridge converter with coupled inductor," *IEEE Trans. Ind. Electron.*, 2025.
- [12] G. Nethaji and J. Kathirvelan, "Performance analysis of FPGA-based semi-dual active bridge DC-DC boost converter using artificial intelligence optimization algorithms," *J. Appl. Math.*, vol. 2025, no. 1, p. 5513346, 2025.
- [13] X. Wei et al., "Evaluation of transformer model and self-attention mechanism in the Yangtze River basin runoff prediction," *J. Hydrol.: Reg. Stud.*, vol. 47, p. 101438, 2023.
- [14] S. Kumar and A. Solanki, "An abstractive text summarization technique using transformer model with self-attention mechanism," *Neural Comput. Appl.*, vol. 35, no. 25, pp. 18603-18622, 2023.
- [15] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [16] R. Hosseinzadeh and M. Sadeghzadeh, "Attention mechanisms in transformers: A general survey," *J. AI Data Min.*, vol. 13, no. 3, pp. 359-368, 2025.
- [17] B. Ghoghoj and A. Ghodsi, "Attention mechanism, transformers, BERT, and GPT: Tutorial and survey," 2020.
- [18] S. Khan et al., "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1-41, 2022.
- [19] D. Soydaner, "Attention mechanism in neural networks: Where it comes and where it goes," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13371-13385, 2022.
- [20] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6688-6697.
- [21] Z. Wu, T. Li, K. Qin, and D. Ruan, "Approximation operators, binary relation and basis algebra in L-fuzzy rough sets," *Fundamenta Informaticae*, vol. 111, no. 1, pp. 47-63, 2011.

- [22] J. M. Davoren et al., “Topological semantics for intuitionistic modal logics, and spatial discretisation by A/D maps,” in Workshop Intuitionistic Modal Logic Appl. (IMLA), Copenhagen, Denmark, Jul. 2002.
- [23] T. Suzuki, “Duality-sealed variation,” in Mean Field Theories and Dual Variation, Paris, France: Atlantis Press, 2009, pp. 1 - 140.
- [24] Y. Wang et al., “Anchor DETR: Query design for transformer-based detector,” in Proc. AAAI Conf. Artif. Intell., vol. 36, no. 3, Jun. 2022, pp. 2567-2575.
- [25] C. Park and Y. Ko, “QSG transformer: Transformer with query-attentive semantic graph for query-focused summarization,” in Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, Jul. 2022, pp. 2589-2594.
- [26] A. P. Bhopale and A. Tiwari, “Transformer-based contextual text representation framework for intelligent information retrieval,” Expert Syst. Appl., vol. 238, p. 121629, 2024.
- [27] J. Wang et al., “SDCDNet: A semi-dual change detection network framework with super-weak label for remote sensing image,” IEEE Trans. Geosci. Remote Sens., vol. 61, pp. 1-14, 2023.
- [28] J. Choi, J. Choi, and M. Kang, “Generative modeling through the semi-dual formulation of unbalanced optimal transport,” in Adv. Neural Inf. Process. Syst., vol. 36, pp. 42433-42455, 2023.
- [29] A. Vacher and F. X. Vialard, “Semi-dual unbalanced quadratic optimal transport: Fast statistical rates and convergent algorithm,” in Proc. Int. Conf. Mach. Learn. (ICML), Jul. 2023, pp. 34734-34758.
- [30] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, “Stochastic optimization for large-scale optimal transport,” in Adv. Neural Inf. Process. Syst., vol. 29, 2016.

APPENDIX:

The complete code for this research experiment:

```
# =====
# Semi-Dual Attention (SDA) vs Standard Transformer
# Semi-Duality Theory Inspired Attention
# PyTorch + Hugging Face datasets, WikiText-2
# RTX 2060 12GB compatible - CAUSAL LM VERSION
# =====

import math

import time

import torch

import torch.nn as nn

import torch.nn.functional as F

import torch.optim as optim

from datasets import load_dataset

from torch.utils.data import DataLoader

from transformers import AutoTokenizer

# -----

# Device

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

torch.manual_seed(42)

# -----

# Load WikiText-2

dataset = load_dataset("wikitext", "wikitext-2-raw-v1")

tokenizer = AutoTokenizer.from_pretrained("gpt2")

tokenizer.pad_token = tokenizer.eos_token

def encode(example):

    return tokenizer(
```

```

        example["text"],
        truncation=True,
        max_length=128,
        padding="max_length"
    )

train_data = dataset["train"].map(encode, batched=True)
val_data = dataset["validation"].map(encode, batched=True)
columns = ["input_ids", "attention_mask"]
train_data.set_format(type="torch", columns=columns)
val_data.set_format(type="torch", columns=columns)
train_loader = DataLoader(train_data, batch_size=16, shuffle=True)
val_loader = DataLoader(val_data, batch_size=16)

# =====
# Standard Multi-Head Attention (Causal)
# =====

class StandardAttention(nn.Module):
    def __init__(self, d_model, n_heads, dropout=0.1):
        super().__init__()
        self.attn = nn.MultiheadAttention(
            d_model, n_heads, dropout=dropout, batch_first=False
        )
    def forward(self, x):
        seq_len = x.size(0)
        causal_mask = torch.triu(
            torch.ones(seq_len, seq_len, device=x.device) * float("-inf"),
            diagonal=1
        )
        out, _ = self.attn(x, x, x, attn_mask=causal_mask)
        return out

# =====
# Semi-Dual Attention (NEW)
# =====

class SemiDualAttention(nn.Module):
    """
    Semi-Dual Attention:
    - Queries are optimized explicitly (primal)
    - Keys act as implicit dual variables
    - Dual feasibility enforced via semi-dual envelope
    """
    def __init__(self, d_model, n_heads, dropout=0.1, alpha=0.5):
        super().__init__()
        self.n_heads = n_heads

```

```

self.d_model = d_model
self.head_dim = d_model // n_heads
assert self.head_dim * n_heads == d_model
self.qkv_proj = nn.Linear(d_model, 3 * d_model)
self.out_proj = nn.Linear(d_model, d_model)
self.dropout = nn.Dropout(dropout)
self.alpha = alpha
def forward(self, x):
    seq_len, batch_size, _ = x.size()
    qkv = self.qkv_proj(x)
    q, k, v = qkv.chunk(3, dim=-1)
    q = q.view(seq_len, batch_size, self.n_heads, self.head_dim).transpose(0, 1).transpose(1, 2)
    k = k.view(seq_len, batch_size, self.n_heads, self.head_dim).transpose(0, 1).transpose(1, 2)
    v = v.view(seq_len, batch_size, self.n_heads, self.head_dim).transpose(0, 1).transpose(1, 2)
    # Raw attention scores
    scores = torch.matmul(q, k.transpose(-2, -1)) / math.sqrt(self.head_dim)
    # -----
    # Semi-Dual Envelope (CORE INNOVATION)
    # -----
    # Dual variable = implicit maximization over keys
    dual_envelope = scores.max(dim=-1, keepdim=True).values
    # Semi-dual constrained score
    semi_dual_scores = scores - self.alpha * dual_envelope
    # Causal Mask
    causal_mask = torch.tril(
        torch.ones(seq_len, seq_len, device=x.device)
    ).view(1, 1, seq_len, seq_len)
    semi_dual_scores = semi_dual_scores.masked_fill(
        causal_mask == 0, float("-inf")
    )
    attn_weights = F.softmax(semi_dual_scores, dim=-1)
    attn_weights = self.dropout(attn_weights)
    out = torch.matmul(attn_weights, v)
    out = out.transpose(1, 2).transpose(0, 1).contiguous()
    out = out.view(seq_len, batch_size, self.d_model)
    return self.out_proj(out)
# =====
# Transformer Block
# =====
class TransformerBlock(nn.Module):
    def __init__(self, d_model, n_heads, d_ff, attention_type="standard"):
        super().__init__()

```

```

if attention_type == "standard":
    self.attn = StandardAttention(d_model, n_heads)
else:
    self.attn = SemiDualAttention(d_model, n_heads)
self.norm1 = nn.LayerNorm(d_model)
self.ff = nn.Sequential(
    nn.Linear(d_model, d_ff),
    nn.ReLU(),
    nn.Linear(d_ff, d_model)
)
self.norm2 = nn.LayerNorm(d_model)
def forward(self, x):
    x = x + self.attn(self.norm1(x))
    x = x + self.ff(self.norm2(x))
    return x

# =====
# Transformer Model
# =====

class SimpleTransformer(nn.Module):
    def __init__(
        self,
        vocab_size,
        d_model=128,
        n_heads=4,
        d_ff=512,
        n_layers=2,
        attention_type="standard"
    ):
        super().__init__()
        self.embedding = nn.Embedding(vocab_size, d_model)
        self.layers = nn.ModuleList([
            TransformerBlock(d_model, n_heads, d_ff, attention_type)
            for _ in range(n_layers)
        ])
        self.fc_out = nn.Linear(d_model, vocab_size)
    def forward(self, x):
        x = x.transpose(0, 1)
        x = self.embedding(x)
        for layer in self.layers:
            x = layer(x)
        x = self.fc_out(x)
        return x.transpose(0, 1)

```

```

# =====
# Training & Evaluation
# =====

def train_epoch(model, loader, optimizer, criterion):
    model.train()
    total_loss = 0.0
    start_time = time.time()
    for batch in loader:
        input_ids = batch["input_ids"].to(device)
        inputs = input_ids[:, :-1]
        targets = input_ids[:, 1:]
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(
            outputs.reshape(-1, outputs.size(-1)),
            targets.reshape(-1)
        )
        loss.backward()
        torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
        optimizer.step()
        total_loss += loss.item()
    elapsed = time.time() - start_time
    avg_loss = total_loss / len(loader)
    ppl = math.exp(avg_loss) if avg_loss < 20 else float("inf")
    return avg_loss, elapsed, ppl

def eval_epoch(model, loader, criterion):
    model.eval()
    total_loss = 0.0
    start_time = time.time()
    with torch.no_grad():
        for batch in loader:
            input_ids = batch["input_ids"].to(device)
            inputs = input_ids[:, :-1]
            targets = input_ids[:, 1:]
            outputs = model(inputs)
            loss = criterion(
                outputs.reshape(-1, outputs.size(-1)),
                targets.reshape(-1)
            )
            total_loss += loss.item()
    elapsed = time.time() - start_time
    avg_loss = total_loss / len(loader)

```

```

    ppl = math.exp(avg_loss) if avg_loss < 20 else float("inf")
    return avg_loss, elapsed, ppl

# =====
# Execution
# =====

vocab_size = tokenizer.vocab_size
num_epochs = 2
lr = 1e-3
criterion = nn.CrossEntropyLoss(ignore_index=tokenizer.pad_token_id)
# -----
# 1. Semi-Dual Attention (Experiment Group)
# -----

print("=== Training Semi-Dual Attention Transformer (Experiment Group) ===")
sda_model = SimpleTransformer(
    vocab_size,
    attention_type="semi_dual"
).to(device)
optimizer = optim.Adam(sda_model.parameters(), lr=lr)
for epoch in range(num_epochs):
    tr_loss, tr_time, tr_ppl = train_epoch(sda_model, train_loader, optimizer, criterion)
    val_loss, val_time, val_ppl = eval_epoch(sda_model, val_loader, criterion)
    print(
        f"Epoch {epoch+1}: "
        f"Train Loss={tr_loss:.4f}, "
        f"Val Loss={val_loss:.4f}, "
        f"PPL={val_ppl:.2f}, "
        f"Time={tr_time:.2f}s (val {val_time:.2f}s)"
    )
# -----
# 2. Standard Attention (Control Group)
# -----

print("\n=== Training Standard Transformer (Control Group) ===")
std_model = SimpleTransformer(
    vocab_size,
    attention_type="standard"
).to(device)
optimizer = optim.Adam(std_model.parameters(), lr=lr)
for epoch in range(num_epochs):
    tr_loss, tr_time, tr_ppl = train_epoch(std_model, train_loader, optimizer, criterion)
    val_loss, val_time, val_ppl = eval_epoch(std_model, val_loader, criterion)
    print(
        f"Epoch {epoch+1}: "

```

```

        f"Train Loss={tr_loss:.4f}, "
        f"Val Loss={val_loss:.4f}, "
        f"PPL={val_ppl:.2f}, "
        f"Time={tr_time:.2f}s (val {val_time:.2f}s)"
    )
print("\n=== Experiment Completed ===")

.to(device)

optimizer = optim.Adam(dioa_model.parameters(), lr=lr)
for epoch in range(num_epochs):
    tr_loss, tr_time, tr_ppl = train_epoch(dioa_model, train_loader, optimizer, criterion)
    val_loss, val_time, val_ppl = eval_epoch(dioa_model, val_loader, criterion)
    print(f"Epoch {epoch+1}: Train Loss={tr_loss:.4f}, Val Loss={val_loss:.4f}, "
          f"PPL={val_ppl:.2f}, Time={tr_time:.2f}s (val: {val_time:.2f}s)")

# 2. Control Group: Standard Transformer
print("\n=== Training Standard Transformer (Control Group) ===")

std_model = SimpleTransformer(vocab_size, attention_type="standard").to(device)
optimizer = optim.Adam(std_model.parameters(), lr=lr)
for epoch in range(num_epochs):
    tr_loss, tr_time, tr_ppl = train_epoch(std_model, train_loader, optimizer, criterion)
    val_loss, val_time, val_ppl = eval_epoch(std_model, val_loader, criterion)
    print(f"Epoch {epoch+1}: Train Loss={tr_loss:.4f}, Val Loss={val_loss:.4f}, "
          f"PPL={val_ppl:.2f}, Time={tr_time:.2f}s (val: {val_time:.2f}s)")
print("=== Experiment Completed ===")

```