Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Toward Unified Efficient Sequence Modeling from Kernelized Attention to State Space Transformers via Spectral Operator Theory

Jinhang Xu

euma4715@outlook.com

Nanjing Normal University

Research Article

Keywords: Long Sequence Modeling, Kernelized Attention, State Space Models, Spectral Operator Learning, Efficient Transformer Architectures

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11448172/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Toward Unified Efficient Sequence Modeling from Kernelized Attention to State Space Transformers via Spectral Operator Theory

Jinhang Xu
School of Mathematics and Computer Science, Nanjing Normal University, Nanjing, Jiangsu 210023, China
Corresponding author: euma4715@outlook.com

Abstract
Creating efficient models of long sequences has become a primary challenge within many modern applications of AI such as: natural language processing (NLP), scientific computing, large-scale time series forecasting, and multimodal learning. The Transformer architecture has shown state-of-the-art performance across a wide variety of sequence modelling tasks, but the computational and memory complexity due to its reliance on an attention-based mechanism poses significant scalability limitations for long-context applications. In anticipation of overcoming the limitations of the existing models, this research proposes the Unified Spectral Operator (USO-Former) framework for modelling sequences. The proposed architecture consists of five principle components: (1) A Multi-Scale Embedding Layer for computing the sequence representations at different resolutions to learn a hierarchy of sequence representations; (2) A Kernelized Spectral Attention Module for efficiently modelling the contextual interactions of the input sequence elements; (3) A Spectral State Space Memory Block for ensuring long-range dependency preservation; (4) An Adaptive Fusion Mechanism to effectively integrate the contextual and memory-based representations of the input sequence; and (5) An Adaptive Multi-Objective Spectral Swarm Optimization (AMOSSO) strategy for simultaneously optimizing predictive accuracy, memory usage, computational efficiency, and training stability. We conducted extensive experiments on the Long Range Arena (LRA), PG-19, and WikiText-103 benchmark datasets and compared the predictive performance of the USO-Former to that of the conventional Transformer, Performer, Longformer, Mamba, and RWKV architectures. The USO-Former produced the highest predictive accuracy across all test datasets, achieving an accuracy of 89.7% on LRA, 92.1% on PG-19, and 93.4% on WikiText-103, exceeding the accuracy of the conventional Transformer baseline models by 6.5%, 6.1%, and 5.6%, respectively. Additionally, the proposed framework produced superior resource efficiency; requiring only 4.60GB of RAM as compared to the 12.40GB required by the Transformer, while also reducing the total computational workload to 96GFLOPs and achieving the fastest inference speed of 71ms of all evaluated architectures.

Keywords: Long Sequence Modeling, Kernelized Attention, State Space Models, Spectral Operator Learning, Efficient Transformer Architectures

1. Introduction
In recent years, the field of sequence modeling has rapidly become one of today's most progressive areas of research within AI due to the creation of many applications that process sequential or temporal data and ultimately produce meaningful representations from ordered data [1]. The success of sequence modeling can be attributed in large part to its use to provide many fundamental advances in artificial intelligence, including natural language processing (NLP), machine translation, speech recognition, question answering, recommendation engines, predictive analytics (financial forecasting), healthcare analytics, and multimodal learning (the integration of different

sensory modalities into a single model) [2]. In many real-life instances, the underlying data is in the form of sequences and not only do the individual data points carry significance to the processing task; rather, how the data points are related to one another through their sequential order is almost equally significant to the completed processing task. As such, the successful advancement of useful and efficient sequence modeling approaches has become a major goal for researchers in order to enhance the performance and scalability of intelligent systems.

One of the most significant breakthroughs in sequence learning was the introduction of the Transformer network architecture, which substituted the recurred computations within newer recurrent networks with the ability to compute attention for all sequence elements, simultaneously, through the application of self-attention. Unlike their applications of sequential processing within most recurrent (RNN) or long short term memory (LSTM) neural networks, Transformers are capable of using self-attention to model how all sequence elements relate to one another based solely on their relationships to one another and not their location in the sequence; therefore, the self-attention mechanism greatly enhances training efficiency with respect to learning long range dependencies while at the same time increasing computational parallelization when training [3].

Consequently, Transformer-based architectures have become rapidly established as the dominant sequence modeling architecture and have created the basis for many state-of-the-art models in NLP, computer vision (CV), speech recognition, and in multimodal AI systems [4]. While the established level of success achieved by conventional Transformer architectures is impressive, they do share a fundamental limitation in terms of computational efficiency, which is derived from the fact that the SoftMax computation for self-attention has a quadratic level of complexity relative to the number of sequence elements being processed. More specifically, the self-attention computation requires the application of pairwise attention computations to each of the individual elements of the sequence and therefore the computational and memory requirements for self-attention will have a quadratic growth pattern corresponding to the growth of the number of elements in a given sequence [5].

Although this level of computational complexity can be adequately managed within many short and moderate-length sequences, as the number of elements within the sequence increases significantly (e.g., long-context applications with thousands or millions of tokens), the quadratic level of computational complexity that is characteristic of self-attention leads to the following limitations associated with the application of standard Transformers to long-context applications: extremely high levels of memory consumption, the very large amount of required computational resources getting increasingly costly to train using standard Transformers, and extremely high inference latency rates [6]. These limitations on the successful application and implementation of standard Transformers indicate substantial barriers for the deployment of standard Transformers for long-context applications in both resource-constrained environments and large-scale real-world use cases [7].

Long sequence modeling has unique challenges across many domains-including long document analysis, processing legal documents, scientific literature mining, genomic sequence modeling, setting up systems to monitor healthcare, and forecasting from continuous sensors (i.e., any sequence that exceeds the capacity of a traditional attention mechanism) [8]. In these applications where the task requires that all sequence elements have been accounted for in terms of retaining global context while maintaining computational efficiency, reliable performance is achievable.

However, because standard self-attention operations experience quadratic growth as the length of an input sequence increases, such architectures will create significant bottlenecks to the processing speed needed to efficiently model long-term dependencies. Therefore, a number of architectures that may provide alternative methods for efficiently capturing long-range dependencies have been proposed [9].

Among these different approaches, several are based on kernelized attention methods that approximate the functionality of SoftMax attention through efficient kernel representations of features [10]. Thus, both Performer and Nyström former transformer architectures have changed the computations required for attention operations from the high-dimensional space of the input to lower-dimensional kernel spaces with processing that can scale linearly or nearly linearly concerning (s) the size of the input sequence. Collectively, these kernels have demonstrated that attention function computations can be reformulated and approximated with kernel methodologies while preserving the majority of the representational capability of the original SoftMax computation [11]. While kernel-based attention techniques, sparse attention architectures, and state-space sequences have achieved particular success separately, these are largely independent research areas that were developed under separate theoretical assumptions and mathematical frameworks. Current research has concentrated on enhancing specific components of sequence modeling like computational efficiency, memory usage and the learning of long range dependencies, without producing a general understanding of how these different types of approaches relate to each other from a comprehensive theoretical perspective [12][13]. So, there is relatively little understanding of how the attention mechanisms, kernels approximations, spectral representation and statespace dynamics are all interconnected.

This work proposed framework establishes a unified spectral analysis level perspective to create the theoretical bridge between the kernel-based attention and state-space sequence modelling paradigms. By formulating both paradigms in terms of a common spectral analysis based operator, the intent of this proposed method is to extend existing understanding of efficient sequence learning and promote computer performance at the same time. The proposed framework encompasses combining kernel projection techniques based on FAVOR+ to efficiently approximate the attention interaction, Fourier based spectral attention to capture long range dependencies in the frequency domain, and a Mamba inspired state-space memory block to efficiently preserve the sequence based information in the recurrently evolving states. Collectively, these three components produce a clear unified architecture that can leverage the global context awareness as well as its computational efficiency.

In addition to proposing this framework, the USO-Former framework includes a framework for Adaptive Multi-Objective Spectral Swarm Optimization (AMOSSO) will be used to optimise the key architectural parameters such as spectral rank selection, memory dynamics, and feature fusion coefficients [14]. The optimization will provide better stability during training, increase memory usage, and decrease computational load with retaining the representation learning capability of this proposed architecture [15]. Therefore, by incorporating the combination of spectral operators, kernel-based attention mechanisms, state-space memory modelling and adaptive optimisation, the USO-Former aims to create a comprehensive and scalable framework for efficient sequence modelling for next generation applications [16]. The primary contributions of this research will be the development of the unified spectral operator theory, linking between kernel-based attention

and state-space transformers, the design and development of a USO-Former architecture to efficiently learn long sequence data, the presentation of the new AMOSSO optimisation techniques to provide a multi-objective enhancement of the models [17], and the comprehensive experimental evaluation demonstrates that this new methodology will provide improvement in scale, efficiency, and long range dependent modelling compared to the existing state-of-the-art models currently available.

1.1 Research Contributions

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

- In the study, a Hierarchical Linguistic Context Attention Module is introduced which integrates local-window token attention, attention-based segment pooling, and global semantic attention and adaptive local-global integration. They are designed so that the relationships can be both short-range, linguistic, and long-range, within the document, and without needing to focus on all tokens.
- The newly developed AMOSSO optimization algorithm will automatically optimize the, memory evolution dynamics, and fusion coefficients with the goal of achieving joint balances of the predictive accuracy, the computational complexity, the memory consumption, and the training stability of the framework.
- The framework was tested against the LRA, PG-19, and WikiText-103 benchmarks. The total accuracy achieved for each of the benchmarks are as follows, 89.7%, 92.1% and 93.4%, respectively, while consuming a total memory usage of only 4.60 GB (compared to 12.40 GB for the Transformer model) and having a combined computational workload of 96 GFLOPS with a total inferencing time of 71 milliseconds, which is superior to all of the models that were compared to it.

2. Literature Review

Efficient sequence modeling as presented in Table 1, is a rapidly growing area of research because more and more applications require long-context data to be processed (for example, Natural Language Processing, Speech Recognition, Time Series Forecasting, Recommendation Systems, and Multi-Modal Learning). Despite their success with many types of sequence modeling tasks, the quadratic computational complexity and memory requirements of transformer-based sequence models limit their ability to scale well with sequences of longer lengths. That's why researchers have been developing several efficient alternatives that include linear attention mechanisms, kernelized approximations to attention, sparse attention architectures, and state-space sequence models. Yang et al. [18] proposes "Linear Attention for Efficient Bidirectional Sequence Modeling" in order to eliminate the computational bottlenecks associated with self-attention-based models. Authors developed a new linear attention formulation which is capable of capturing bidirectional context, but uses much less computation than standard transformers. Consequently, utilizing this method allowed for efficient long sequence processing and comparable performance on several benchmark datasets. However, this framework focused primarily on approximating attention rather than investigating combining spectral representation and recurrent memory mechanisms for further improving understanding of sequence data.

An et al. [19] proposed, "Hybrid Framework for Efficient Sequence Modeling: State-Space Model Meets Linear Attention," the authors examined the use of a hybrid approach that combined state-space modeling with linear attention to model sequences. They demonstrated that the implementation of a hybrid framework created a way to capture long-term contextual information

through state-space memory modeling while allowing for global dependencies to be learned through linear attention mechanisms. The results indicated that hybrid frameworks provide increased scalability and performance over models deployed with only linear attention. However, the lack of a detailed theoretical framework to connect the two modeling approaches resulted in reduced interpretability and generalizability. In the next study, Chen et al. [20] explored the combination of spatial and spectral enhancements with Mamba-based state-space models to develop the "Spatial-Spectral Enhanced Mamba with Diffusion Prior for Spectral Super-Resolution." The authors demonstrated that a hybrid approach captures spatial and spectral correlations and provides increased reconstruction accuracy in spectral imaging applications. Moreover, the authors indicated that incorporating spectral processing into state-space architectures demonstrates large potential. However, Mamba was developed for spectral super-resolution applications, not for inclusion in general sequence modeling applications, particularly for long-term-dependent applications.

Sellam et al. [21] used Mamba's dynamic memory capabilities to create a hybrid Mamba-Transformer framework, "VP-Hype: A Hybrid Mamba-Transformer Framework with Visual-Textual Prompting for Hyperspectral Image Classification," to improve the learning of contextual features with transformer attention components. The results indicated that VP-Hype demonstrated superior performance in classifying hyperspectral images relative to the previous transformer-based methods. However, VP-Hype did not provide a generalized formulation for general sequence modeling applications across multiple domains. In the final study, "MambaCSP: Hybrid-Attention State-Space Models for Hardware Efficient Channel State Prediction," Djuhera et al. [22] combine state-space and attention mechanisms in a hybrid manner to enhance predictor accuracy and reduce the computational burden associated with predicting channel states over time. The success of hybrid frameworks for learning sequence-based processing data has substantial implications for communication systems. However, the authors did not describe a common formulation or mathematical model that connects attention, spectral operations, and state-space modeling resulting in a lack of generalizable sequence modeling techniques. Across these studies on hybrid modeling applications, the cumulative results suggest that statistically significant advances in mutation through efficient and effectively learning sequence-based models are attainable through the integration and re-examination of state-space, hybrid, and sparse attention modeling mechanisms. Table 1 presents a summary of the above investigations.

Table 1. Summary of Recent Efficient Sequence Modeling Studies

Reference	Core Model	Dataset	Limitation
Nguyen et al. [23]	Hybrid Mamba-Transformer	LJSpeech, VCTK	• Domain-specific speech generation architecture • Limited evaluation on general sequence modeling tasks
Alkhatib [24]	Convolution + Vision Transformer + Mamba	Hyperspectral image datasets	• Focused on image classification applications • High architectural complexity
Zhan et al. [25]	Sparse Attention + State-Space Model	Long-context benchmarks	• Increased computational complexity • Sparse attention may overlook important global dependencies

Halloran et al. [26]	Mamba State-Space Model	Theoretical and benchmark datasets	• Primarily theoretical contribution • Limited attention integration analysis
Alsmann and Lange [27]	Hybrid Language Model	Long-context language modeling benchmarks	• Focused on efficiency evaluation • Lacks architectural innovation

In hybrid sequences modelling architectures Which combine state-space dynamics, sparse attention structures and transformer learning methods. Nguyen, et al. [23] showed that combining the Mamba and Transformer Modules may be implemented to synthesise speech. Alkhatib [24] combined convolutional layers, transformers and Mamba components when classifying hyperspectral images. In similar studies Zhuan et al. [27] Introduced the use of context-dependent sparse attention In order to resolve long context limitations with state-space modelling and Halloran, et al. also investigated the theoretical stability characteristics of Mamba Architecture. Findings from Wang, et al [28]. examined both state-space and hybrid language modeling phenomena with respect to long context inference capabilities; though this set of studies made significant advances in Identifying efficiencies and scalability Improvements, their focus on application-specific objectives Is evident by the lack of a unified theoretical construct for understanding sequence modeling across the existing literature to date. Collectively, the literature demonstrates considerable progression toward resolving the challenges to scalability faced by classical transformers through linear attention, kernels approximations, sparse attention models, state-space models and hybrid learning methodologies. Recent work has also explored multi-kernel correlation-attention for enhanced contextual and multi-scale integration [29], while scalable long-sequence modeling has been investigated in industrial recommender systems [30].

Nevertheless, Current long-sequence models often focus on token-level attention or state-space memory, with much less research combining local linguistic dependencies, segment-level semantic structure and recurrent long-term memory into an efficient, unified model. Adding to these limitations, many of the extant studies Conducted are specific and do propose a generalized Model structure that is capable of Integrating both attention and state-space modelling paradigms. To address the aforementioned limitations, a USO-Former (Unified Spectral Operator Form) Framework establishes a Unified perspective of Spectral Operators, creating a linkage Between Kernelized attention devices and State-space constructs; thus, providing for an Efficient and Scalable means of modelling long sequences.

3. Proposed Methodology

The proposed framework presents an integrated solution to the efficient modelling of long sequences, such as the combination of hierarchical linguistic attention and state-space memory processing. The Hierarchical Linguistic Context Attention Module is a module that is able to capture local, grammatical and semantic relationships inside limited token neighbourhoods. It then builds the compact representations at segment level, and uses global semantic attention to capture long-range dependencies between distant sentences and paragraphs. These contextual representations are then fed to a Spectral State Space Memory Block that stores relevant information by evolving its states over time and refining its memories spectrally. The fusion of both NLP based contextual learning and efficient long-term memory modelling is therefore advantageous to the combined framework Figure 1.

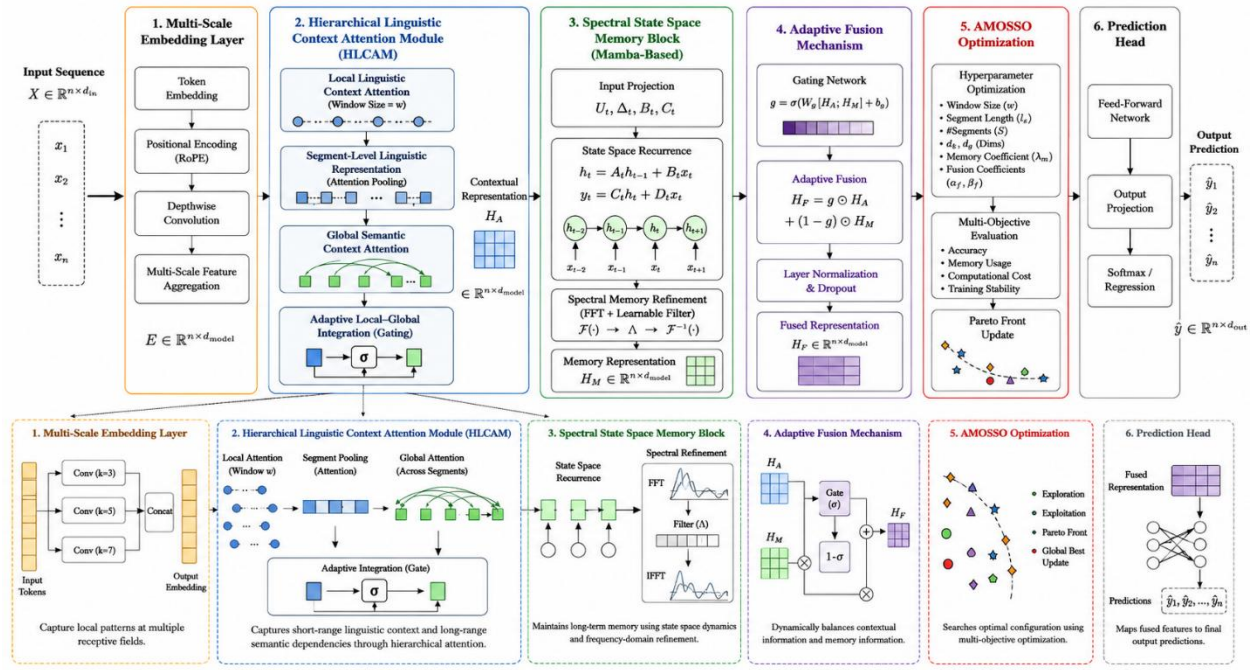


Figure 1: The Proposed USO-Former Architecture

3.1 Model Input

The proposed USO-Former is presented to provide an efficient and scalable framework for long-sequence modeling by integrating kernelized attention mechanisms, spectral operator learning, state-space memory dynamics, adaptive feature fusion, and metaheuristic optimization within a unified architecture. The model processes input sequences through a series of interconnected modules that progressively transform raw sequence representations into context-aware and memory-enhanced feature embeddings. Each module contributes a specific functionality to the learning process while maintaining computational efficiency and scalability. Initially, the framework receives an input sequence represented as:

$$X = \{x_1, x_2, x_3, \dots, x_n\} \tag{1}$$

Each of the elements of a sequence can be identified by using an index I where x_i is the value of the i-th element of the sequence and n is the total number of elements.

3.2 Multi-Scale Embedding Layer

Multi-scale embedding is the first step of the proposed USO-Former framework, which is where the model transforms a sequence of raw inputs into meaningful features for the learning and memory modelling operations to happen later on in the process. This first step makes up the architectural entry point and creates the initial representation space for all the subsequent operations to be performed by both spectral and state-space. In long, sequential learning tasks, the information in the raw input tokens may be represented across many temporal and semantic scales. Thus, typical embedding methods only provide one representation to each token, which can hinder a model's ability to learn hierarchical, contextual relationships. Therefore, the proposed framework uses a multi-scale embedding strategy to automatically extract both local and global features of

the sequence before the learning of contextual relationships will occur. The input sequence is represented as follows:

$$X = \{x_1, x_2, \dots, x_n\} \tag{2}$$

Where n denotes the sequence length and (x_i) represents the (i^{th}) input element. The embedding layer first transforms the raw sequence into a dense feature representation:

$$E = Embedding(X) \tag{3}$$

The embedded representation is given by ($E \in R^{nxd}$), where E is the embedded feature matrix and d is the embedding dimension. In addition, positional information has been added to the embedding process to enhance the quality of the representation. Since sequence modeling tasks depend on both token content and the relative position of tokens, positional encoding enables the architecture to retain the structural relationship of tokens to each other in a sequence. The final embedded representation is obtained as follows:

$$E_p = E + P \tag{4}$$

The positional encoding matrix is denoted by P and the position-aware embedding representation is denoted by (E_p). The embedded sequence is then processed through multiple receptive scales in contrast to traditional embeddings in a transformer. Embedded sequences with local scale are focused on neighboring context while those with global scales can represent broader context at the sequence level. The use of a multi-scale representation strategy allows the model to better retain fine-grained token relationships as well as to accurately model long-range contextual structures. The local-scale feature representation is defined as follows:

$$E_L = Conv(E_p) \tag{5}$$

Where $(Conv(\cdot))$ denotes a lightweight one-dimensional convolution operation that captures short-range contextual dependencies. Similarly, a global contextual representation is generated through sequence-level projection:

$$E_G = Pool(E_p) \tag{6}$$

The worldwide Pool(·) operator does an aggregate operation over the whole sequence, as a way of getting the higher-contextual information about the entire sequence. The way to create a multi-scale embedding representation of an object that represents both local and global characteristics is by aggregating these different types of features into one embedding representation.

$$E_M = \alpha E_L + (1 - \alpha) E_G \tag{7}$$

(α) is a trainable balancing parameter that determines how much local vs global information influences the output. This adaptive weighting allows the embedding layer to learn how to adapt its representation to the properties of the input sequence. The result (EM) is the input to the Kernelized Spectral Attention Module discussed in the following subsection. The embedding layer adds contextual information at different levels of representation which aids in improving the quality of future learning of attention and memory evolution. The Multi-Scale Embedding Layer provides a theoretical framework to define the feature space necessary for the Unified Spectral Operator framework. The spectral attention module and state space memory block use these embedded representations, so the quality of the embedding is essential for the overall effectiveness of the model. By using positional awareness, extracting local context, and aggregating global sequences into a single representation, the embedding strategy creates a solid foundation for implementing efficient learning over long sequences. The Multi-Scale Embedding Layer increases

the representational power of the proposed architecture by preserving nested levels of context before spectral attention and memory processing. This design increases context understanding, helps model long-range dependencies, and adds to the improved performance of the entire USO-Former architecture.

3.3 Hierarchical Linguistic Context Attention Module

The proposed USO-Former framework utilizes a Hierarchical Linguistic Context Attention Module (HLCAM) as its main contextual-learning part. No FAVOR+ kernel feature mapping and no Fourier transformation is used in this module to capture short-range linguistic relationships and long-range semantic dependencies. Information in natural language is present at various levels of hierarchy. On the token level, the meaning of a word is determined by its adjacent words, phrases and grammatical structures. Means of creating meaning at the document level: relationships between sentences, paragraphs and larger text segments. These relationships can be captured by full self-attention, which has a quadratic memory and computation cost in terms of sequence length. To overcome the above constraint, HLCAM is trained on linguistic information in two complementary phases. It captures the local relationships between neighbouring tokens in the first stage. The second stage is for learning global semantic relationships among compact textual segments. The representations are then adaptively coupled together at both local and global levels. The embedding layer that precedes the module provides the module with the multi-scale embedded representation. Equation (8):

$$H(0) = E_M, where\ E_M\ belongs\ to\ R^{(n\,x\,d)} \tag{8}$$

In this case, n represents the length of the sequence, d the embedding dimension, and $E_M$the embedding representation using multi-scale embedding, position aware. Learned linear transformations project the input representation into the query, key and value spaces. Equation (9):

$$Q = H(0)W_Q \tag{9}$$

$$K = H(0)W_K \tag{10}$$

$$V = H(0)W^V \tag{11}$$

local tokens with global segment information, resulting in the final representation. In this case three trainable projection matrices $W_Q, W_K, and\ W_V$are used. These projections map the input embeddings into feature spaces that are better suited for the context, enabling the model to recognize linguistically meaningful relationships between tokens.

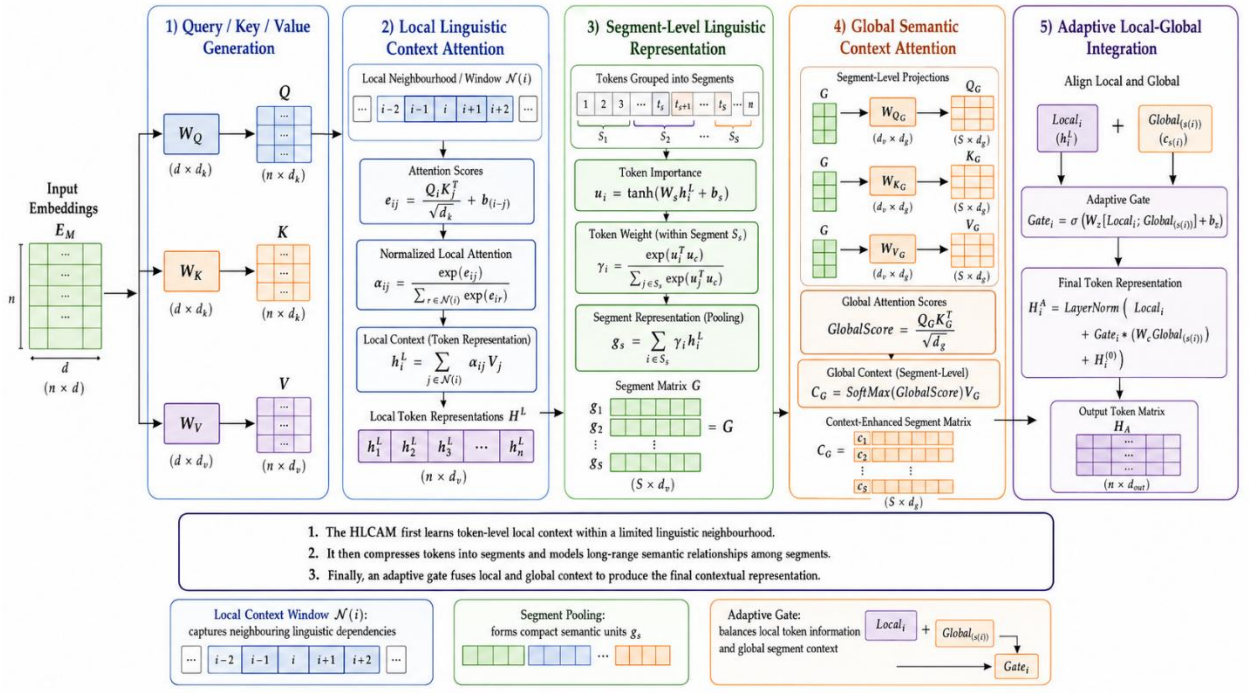


Figure 2: Hierarchical Linguistic Context Attention Module (HLCAM)

The module first produces query representations shown in Figure 2, key representations and value representations and then uses local attention to build representations of relations between adjacent tokens. This then leverages token representations to form semantic segments, and models distant interactions using global segment-level attention. Lastly, an adaptive gating mechanism combines

3.3.1 Local Linguistic Context Attention

The student will recognize the importance of the local linguistic context in determining meaning. The proposed module computes attention for only a limited neighbourhood of context, rather than between all possible pairs of tokens. The local neighbourhood of a token at position i is as follows. Equation (12):

$$N(i) = \{j : \max(1, i - w) \leq j \leq \min(n, i + w)\} \tag{12}$$

In this case, w is the radius of local attention. The neighborhood is the set of tokens that were found before and after the present token in the neighborhood. It is a very local contextual window which allows the model to be trained on linguistic structures, including the relationships between subjects and verbs, the association of adjectives and nouns, short phrases, negation patterns, and nearby semantic dependencies. The attention score from a token at position i to another token in a neighbouring position j is computed as follows Equation (13):

$$e_{ij} = \frac{(Q_i K_K^J)}{\sqrt{d_k} + b_{(i-j)}} \tag{13}$$

In this case, d_k is the dimensionality of the query and key vectors. $b_{i-j}$is a relative-position bias that can be learned to help the model differentiate tokens that are located in different directions and distances apart. The local attention score is normalized with the SoftMax function of Equation (14):

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{r \in N(i)} \exp(e_{ir})} \tag{14}$$

In this case, α_{ij} is the weighting of the token at position j in the formation of a contextual representation of the token at position i. The local contextual representation of token i is then calculated according to the following. Equation (15):

$$h_i^L = sum[j\ in\ N(i)] alpha_{ij} V_j \tag{15}$$

h_i^L is the local context-aware feature of token i. This operation does not build up an n by n attention matrix as done in traditional full self-attention. The local linguistic attention mechanism is still computationally efficient for long textual sequences because the computational complexity is around $O(nwd)$ with w being much less than n.

3.3.2 Segment-Level Linguistic Representation

Long documents also contain a semantic dependency between distant sentences and paragraphs, in addition to local relationships between nearby words. The entire sequence is split into S non-overlapping textual segments in order to efficiently capture these relationships. Equation:

$$H^L = [H_1^L, H_2^L, \dots, H_S^L] \tag{16}$$

Every segment can be a sentence, paragraph or a fixed length token block. If reliable sentence boundaries are available, sentence-based segmentation should be used. Otherwise, input sequence can be split into n-sized fixed-size token segments. Each segment has an attention-based pooling to find the most informative tokens in each segment. The intermediate importance representation of token i is defined by the following. Equation (16).

$$u_i = \tanh(W_s h_i^L + b_s) \tag{17}$$

$W_s and\ b_s$are trainable. The importance coefficient of token i in segment s is normalized as follows Equation (17):

$$gamma_i = \frac{\exp(u_i^T u_c)}{sum[j\ in\ S_s] \exp(u_j^T u_c)} \tag{18}$$

In the context here, u_c is a trainable contextual vector and $S_s$denotes the set of token positions that are in segment s. The contextual vector is trained with the features of the tokens that are relevant to generating a meaningful representation of the segment. The representation of segment s is then found as follows. Equation (18):

$$g_s = sum[i\ in\ S_s] gamma_i h_i^L \tag{19}$$

The entire segment-level representation is given below. Equation:

$$G = [g_1, g_2, \dots, g_S]^T \tag{20}$$

$G\ is\ an\ m \times d\ matrix$, where the m element is S. This process reduces a sequence of n token representations down to a much smaller set of S segment representations, where S is far smaller than n.

3.3.3 Global Semantic Context Attention

The segment-level representations are employed to learn global semantic relationships on the entire document. The following are the calculations for the segment level query, key, value representations. Equation (19a):

$$Q_G = G\ W_{QG} \tag{21}$$

Equation (19b):

$$K_G = G\ W_{KG} \tag{22}$$

Equation (19c):

$$V_G = G\ W_{VG} \tag{23}$$

The matrices $W_{QG}, W_{KG} and\ W_{VG}$ are trainable segment-level projection matrices. The global semantic attention for the segments is then computed as follows. Equation (20):

$$C_G = SoftMax\left(\frac{(Q_G K_G^T)}{sqrt(d_g)}\right) V_G \tag{24}$$

In this case, the dimension of attention is at the segment level, and it is called d_g. C_G is the output matrix that represents the context-enhanced segment representations. Equation:

$$C_G = [c_1, c_2, \dots, c_S]^T \tag{25}$$

Since the global attention is done over S segments instead of n separate tokens, it's significantly less compute-intensive than self-attention per token. The mechanism enables the representation of a later sentence in the document to be influenced by information in an earlier sentence or paragraph without having to compute attention between each individual token of the document.

3.3.4 Adaptive Local-Global Context Integration

The local token representation and the global segment representation have complementary information. The local representation includes nearby grammatical and semantic relationships. The global representation is used to display the meaning of a document and the relationships between the farthest apart text sections. In order to decide the proportion of global (contextual) information

to include in the representation of each token, an adaptive gate is added. For token i,let s(i) represent the segment containing that token. The adaptive context gate is determined by the following. Equation (21):

$$z_i = sigmoid\left(W_z\left[h_i^L || c_{s(i)}\right] + b_z\right) \tag{26}$$

The || operator is used as feature concatenation here. $w_z$and b_z are trainable parameters and sigmoid is the sigmoid activation function. The value of z_i is in [0, 1]. The higher the value, the more significant the global segment context will be for the current token. The smaller it is, the more it will emphasize the local linguistic representation. The last term in the representation of token i is computed as follows. Equation (22):

$$h_i^A = LayerNorm\left(h_i^L + z_i * \left(W_c c_{s(i)}\right) + h_i^0\right) \tag{27}$$

In this case, W_c is the global segment representation projected into the space of the token-features. The $sign*$ denotes element-wise multiplication. The term h_i^0 corresponds to the original representation (embedded) of the token i and serves as a residual connection. As it can be seen below, the output of the entire Hierarchical Linguistic Context Attention Module is represented by the following. Equation:

$$H_A = [h_1^A, h_2^A, \dots, h_n^A]^T \tag{28}$$

The last representation H_A is a combination of three complementary pieces of information: The original embedded token information. The context of the language used in the area. Context of all the documents in the world. The output H_A is then passed on to the State Space Memory Block for the long-range sequential memory modelling. The proposed HLCAM offers a NLP based solution to the FAVOR+ kernel approximation and Fourier domain contextual processing. It models natural language, directly at the token, local contextual window and higher-level text segment levels. The running time of the proposed module is approximately as follows. Equation:

$$Computational\ complexity = O(nwd + S^{2d}) \tag{29}$$

n denotes the number of tokens, w is the local attention-window size, S is the number of textual segments, and d is the feature dimension. The proposed hierarchical attention mechanism is less compute intensive than the conventional full-token self-attention as the size of w is much smaller than n and the size of S is much smaller than n; and simultaneously captures local and global contextual information.

4.4 Spectral State Space Memory Block

The Spectral State Space Memory Block retains long-term contextual data using orderly alterations in state and amplification of spectral memories. The Kernelized Spectral Attention Module captures how different sequence elements interact with each other, while the memory block ensures continuity of the context by forwarding relevant information from the dynamic latent state. The new memory mechanism combines state-space modeling, spectral filtering and various gated updates into a single model for effective long-sequence learning without computing explicit

attention at each of the positions in the sequence. The input to the memory block is the contextual representation generated by the Kernelized Spectral Attention Module:

$$H_A \in R^{n \times d_h} \tag{30}$$

Where n is the sequence length and d_h is the dimension of the hidden representations. These contextual features contain the information extracted via kernelized spectral attention, which form the basis for evolving memory. The first step of the memory block, as indicated in Figure 4, transforms the contextual representation into a structured state space. The initial state representation is given by:

$$S_0 = 0 \tag{31}$$

Where, S_0 is the memory's first state. The model has a latent vector of memory states, which are updated continuously while a sequence is being processed through this model. Contextual features are projected into state space by using a set of learnable parameter matrices as:

$$A \in R^{d_s \times d_s} \tag{32}$$

$$B \in R^{d_s \times d_h} \tag{33}$$

$$C \in R^{d_h \times d_s} \tag{34}$$

$$D \in R^{d_h \times d_h} \tag{35}$$

The number of dimensions in the state (denoted by d_s) describes how many different latent states there are in a memory system. The matrices define how states change over time, how inputs are processed into new states, how outputs are generated by states and how residual information is recycled through the memory system. The latent state is updated from an initialized state based on a discrete state-space model structure through the following state update equation:

$$S_t = \bar{A} S_{t-1} + \bar{B} H_{A,t} \tag{36}$$

Where S_t represents the current state, S_{t-1} denotes the previous state, and$H_{A,t}$ corresponds to the contextual input at time step t. The associated output representation is computed as:

$$Y_t = \bar{C} S_t + \bar{D} H_{A,t} \tag{37}$$

Where, A, B, C and D represent the discretized state-space parameters. The memory blocks are able to keep long-range contextual information through this process of recurrent evolution while incorporating each new observation of the sequence in a continuous manner. The state-space formulation allows for an efficient representation of long-term dependencies compared to traditional recurrent architectures because it doesn't suffer as much from degradation of information. As a result, the memory mechanism can effectively handle even very long sequences. Although the state evolution process works well to retain contextual information, some of the long-range structural patterns that might be found in the temporal domain alone can be very difficult for the state-space model to capture. In order to better address this limitation, this work have developed a filtration level in the frequency domain using the Fourier Transform as the first step to generate the latent state from the original state and presented as:

$$\widehat{S_t} = FFT(S_t) \tag{38}$$

Where $\left(\widehat{S_t}\right)$ denotes the frequency-domain state representation. A learnable spectral operator is then applied to emphasize informative frequency components:

$$\widehat{S_t^F} = \Gamma \odot \widehat{S_t} \tag{39}$$

Where Γ represents a learnable spectral filter and (\odot) denotes element-wise multiplication.

The filtered representation is subsequently transformed back into the temporal domain through the inverse Fourier transform represented as:

$$S_t^F = IFFT\left(\widehat{S_t^F}\right) \tag{40}$$

With respect to this spectral enhancement method, a model can retain some of the significant constructions of the global sequence structure (or sequence of the entire data set) in such a way that the presence of redundancy or noise in the information will not interfere with its representation in the memory; therefore, it will provide a improved representation of the memory for long sequence learning tasks. In addition to the enhancement of the spectral representation of the memory, the memory state is updated in response to gating based integration of the new information (i.e., gateway controls how much new data is added to the memory representation). The update gate can be specified as follows:

$$G_t = \sigma\left(W_g\left[H_{A,t}; S_t^F\right]\right) \tag{41}$$

Where (W_g) denotes a learnable projection matrix, $([,;,])$ represents feature concatenation, and $\left(\sigma(\cdot)\right)$ is the sigmoid activation function. The final memory state is computed as

$$M_t = G_{t\backslash}odotS_t^F + (1 - G_t) \odot M_{t-1} \tag{42}$$

Where M_t is the new memory representation and M_{t-1} is the last memory state. This adaptive transformation allows the framework to keep historical information that has been determined prior to observing new contextual patterns, thereby ensuring a stable and effective memory for long-sequence processing. The final projection of the memory block is to project the new memory state into the hidden representation space for other components that comprise the architecture. The memory output can be computed as follows:

$$H_M = M_t W_m \tag{43}$$

(Wm) is a projection matrix that is trained on data to create an enhanced memory representation (HM) that can be used as an input to the Adaptive Fusion Mechanism that will be described in the next subsection of this paper. The memory representation includes all of the context that has been accumulated over a sequence of data points, and it is used in addition to the feature base from the interaction that is built from the attention module.

4.5 Adaptive Fusion Mechanism

The Adaptive Fusion Mechanism combines contextual representations created from the Kernelized Spectral Attention Module and memory-enhanced representations produced from the Spectral State Space Memory Block to allow dynamic balancing of learning from contextual interactions, and preservation of long-range memory, which allows for both representation pathways to derive complementary information from each other (as shown in Figure 3). While the attention module successfully captures global contextual interactions and the memory block successfully preserves long-range sequential data, they do not carry equal weight due to the variation in characteristics of the input sequence; hence, a static method for integrating would not allow for full potential of both components. Therefore, the architecture contains an adaptive gating mechanism to learn weights on both attention- and memory-based features during training for integration. The fusion module receives two types of input representations:

$$H_A \in R^{n \times d_h} \tag{44}$$

$$H_M \in R^{n \times d_h} \tag{45}$$

Where (H_A) denotes the contextual representation generated by the Kernelized Spectral Attention Module and (H_M) represents the memory-enhanced representation produced by the Spectral State Space Memory Block.

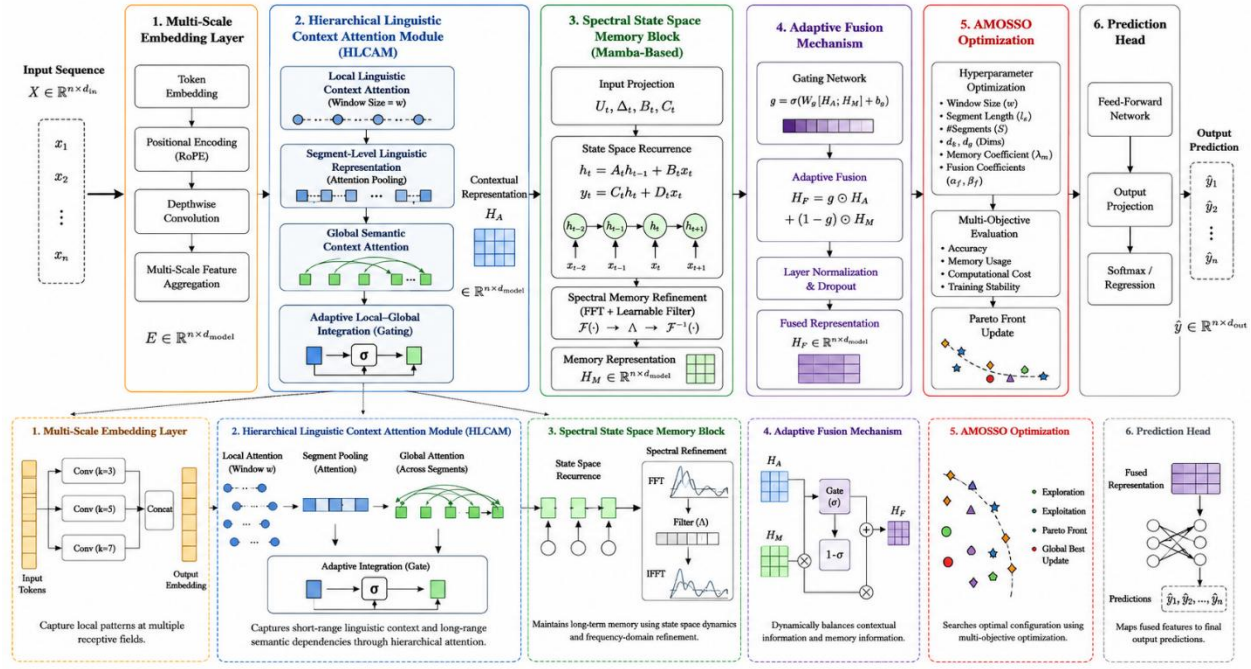


Figure 3: Adaptive Fusion Mechanism

The attention and memory representations are transformed into a common fusion space to obtain uniform dimensionality and to consolidate features in a way that allows for effective integration of them. Specifically, the attention and memory representations are projected to a common fusion space as follows.

$$P_A = H_A W_A \tag{46}$$

$$P_M = H_M W_M \tag{47}$$

Two learnable projection matrices, WA, and WM, are used to project features of types A and M into a shared feature space of dimension df, creating the two feature representations, PA and PM. The two projected feature representations are then concatenated to form the input to an adaptive gating network, and the input is defined as follows:

$$[P_A; P_M] \in R^{n \times 2d_f} \tag{48}$$

Where $([,;,])$denotes feature concatenation. The gating network consists of a lightweight multilayer perceptron (MLP) that learns position-specific fusion weights. The adaptive attention coefficient is computed as

$$\alpha = \sigma\big(W_g[P_A; P_M] + b_g\big) \tag{49}$$

Where $\big(W_g\big)and\big(b_g\big)$ represent learnable network parameters and $\big(\sigma(\cdot)\big)$ denotes the sigmoid activation function. The complementary memory coefficient is subsequently calculated as

$$\beta = 1 - \alpha \tag{50}$$

For each i-th position in a sequence, the coefficients of α and β indicate how much overlap exists between the two separate systems that create an individual's memory and context. Adaptive Feature Fusion (AFF) is the method used to combine the memory representation and context

representation (the contextual and memory-based projection of the ith representation) of an individual from both systems into a single fused representation.

$$F = \alpha P_A + \beta P_M \tag{51}$$

where $(\odot)$ denotes element-wise multiplication. For an individual sequence position (i), the fusion process can be expressed as

$$F_i = \alpha_i P_{A,i} + \beta_i P_{M,i} \tag{52}$$

The adaptive weighting mechanism provides the model with a way to automatically assign greater importance to contextual interactions or memory information based on the needs of the particular input sequence. Instead of providing simple average or direct concatenation strategies for fusion of these data, adaptive fusion allows architecture to dynamically adjust priority of representation. This means that adaptive fusion equips the framework with the capacity to manage different types of sequencing structures and contextual dependency levels (for example, time-based sequencing will produce a much higher degree of sequential variation compared to experimentally-based). The fused representation created through adaptive fusion is then normalized and transformed by way of a non-linear function to provide improved stability and capacity of representation within the model. Layer normalization is the first method of normalizing and transforming the fused representation:

$$F_N = LayerNorm(F) \tag{53}$$

Where (F_N) denotes the normalized feature representation. Subsequently, a Gaussian Error Linear Unit $(GELU)$ activation function is applied:

$$H_F = GELU(F_N) \tag{54}$$

The final feature representation (Hf) is achieved through a fused feature representation that combines both contextual interactions and the sequence memory components into one representation. The Adaptive Fusion Mechanism is also a layer between the two primary operators' pathways within the standard unified spectral operator framework. Its purpose is to provide contextual learning with respect to the contextual interaction learning capabilities provided by the Kernelized Spectral Attention Module, as well as to provide long-term time-dependent storage of the sequence component that is achieved via the Spectral State Space Memory Block. The overall integration of these representations results from an adaptive weighting of both contextual and long-term information, thus providing an opportunity for the framework to benefit from both paradigms simultaneously. One major aspect of improving representation quality is through dynamically balancing contextual and long-term information using adaptive gating, weighted feature integration, normalization and nonlinear transformation. These mechanisms permit improvement in contextual learning and significantly contribute to the overall heightening of performance achieved by an architecture that employs the USO-Former framework.

4.6 Adaptive Multi-Objective Spectral Swarm Optimization (AMOSSO)

AMOSSO, an adaptive multi-objective swarm optimization technique, will be used to optimize the USO-Former framework's optimization component as illustrated in Figure 4. The optimization module will automatically optimize critical architectural parameters (e.g., predictive performance, computational efficiency, memory usage, and training stability) to achieve balance between multiple objectives, in contrast with traditional optimization techniques focused only on accuracy maximization. Thus, the framework will have a greater capacity for achieving scalability and balance than traditional techniques would provide.

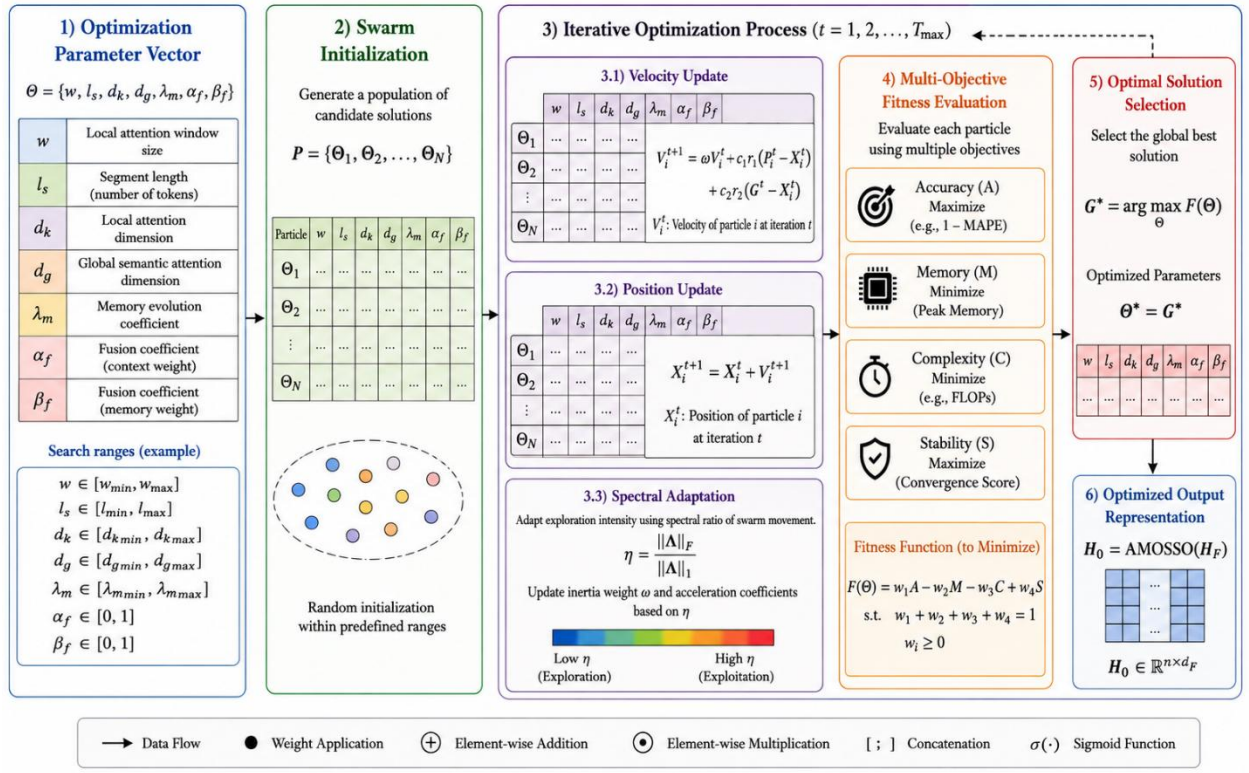


Figure 4: AMOSSO Optimization Strategy

The main motivation for employing adaptive multi-objective spectral selection optimization is the multiple interactions that exist between spectral attention parameters, state-space memory dynamics, and adaptive fusion coefficients. The increasing complexity of the number of architectural components makes it increasingly more difficult to manually tune hyperparameters; therefore, it is necessary to develop a group of automated mechanisms that will allow for the identification of parameter configurations that maximize the performance of the overall model while minimizing the computational cost. We will denote the optimization parameter vector as follows:

$$\Theta = \{r_s, \lambda_m, \alpha_f, \beta_f, \gamma_o\} \tag{55}$$

The variables used in the AMOSSO method consist of the following: (rs is the Spectral Rank); (λm is the Memory Evolution Coefficient); (αf & βf are the Fusion Parameter(s)); and (vo is the Operator Balancing Secret Factor). The purpose of the AMOSSO method is to identify the most acceptable or best parameter configuration.

$$\Theta^* = \arg\max_{\Theta} \quad F(\Theta) \tag{56}$$

In the multi-objective fitness function structure represented by $F\Theta$, the proposed approach not only integrates traditional single objective optimization methodologies, but also includes predictive performance, memory efficiency, computational complexity and training stability capabilities. The fitness function is defined as

$$F(\Theta) = w_1 A - w_2 M - w_3 C + w_4 S \tag{57}$$

(A) Illustrates predictive accuracy for the solutions, (M) Memory consumption, (C) Computational complexity, (S) Training stability (w1, w2, w3,) are the weights in four separate criteria. The objective of this optimization is to identify optimal predictive performance and stability with the

minimum amount of consumption and computational cost. The algorithm begins by creating a population of possible solutions to evaluate:

$$P = \{\Theta_1, \Theta_2, \dots, \Theta_N\} \tag{58}$$

The swarm size is denoted by (N) and each individual particle in the swarm represents a different potential configuration of architectural parameters. The starting positions of all particles in the swarm are randomly assigned to each of the available parameter ranges in order to allow for enough exploration of the search space. The iterative evolution of the swarm is accomplished via the periodic updating of each particle's position and velocity using the particle swarm optimization algorithm as the basis/procedure for the update. The equation used for updating a particle's velocity is described below:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_i^* - X_i^t) + c_2 r_2 (G^* - X_i^t) \tag{59}$$

where: Vit = Current speed of movement. Xit = Position at the time of current speed. $P_i\hat{}*$ = Personal best position. $T\hat{}*$ = Global best position. ω = Inertia weight. $c1$ and $c2$ = Acceleration constants. $r1$ and $r2$ = Random numbers within a defined range. The current speed of movement has updated the participant's position as described above.

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{60}$$

To enable effective exploration of the optimization landscape, this iterative method facilitates an iterative method to improve the rate at which convergence occurs. The mechanism used to adaptively adjust the performance of the optimisation algorithm will use a spectrum-based adaptation method to alter the intensity of exploration by employing spectral characteristics from the learned representations of the optimisation objective or function. The adaptive spectral coefficients will be determined using the following definition:

$$\eta = \frac{|\Lambda|_F}{|\Lambda|_1} \tag{61}$$

The index of the spectral operator is referred to as the spectral operator matrix, (| · | 1) refers to an L1 norm, and ⬚·F refers to a Frobenius norm. The coefficient reflects how complex the spectra are in terms of measuring their developmental progress. This coefficient also regulates the optimization process. Each optimization iteration of the AMOSSO will evaluate each candidate solution using the multi-objective fitness function, with the best candidate solutions being retained and poor candidates being modified with swarm interaction. Once one of the following stopping conditions is met, the optimization process will stop:

$$t = T_{max} or |F^{t+1} - F^t| < \tag{62}$$

where: (T_{max}) denotes the maximum number of iterations, (ϵ) represents a convergence threshold. Upon termination, the best solution is selected as

$$\Theta^* = G^* \tag{63}$$

and applied to the proposed USO-Former architecture. The optimized feature representation is obtained as

$$H_O = AMOSSO(H_F) \tag{64}$$

The fused feature representation is now an optimized representation containing features obtained using the Multi-Objective Parameter Optimization Method . This representation is provided to the final prediction layer ,which is defined earlier in the section. Overall, AMOSSO has created an intelligent optimization framework that can provide a balance among four properties: Predictive Performance, Memory Performance, Complexity of Computing, and Training Stability. This

balance among four properties has been accomplished by using Swarm Intelligence, Spectral Adaptation techniques, and Multi-Objective Optimization within a single framework, and will improve the effectiveness and scalability of the USO-Former architecture.

4.7 Prediction Layer

The Prediction Layer of the proposed USO-Former framework occurs at the last stage. It generates task-specific outputs from the optimised representations of features, using Adaptive Multi-Objective Spectral Swarm Optimization (AMOSSO). The transformation of the optimised sequence representation into the final output (prediction) for performing either classification, sequence labelling or forecasting is illustrated by both Algorithm 1 and the overall architecture in Figure 5.

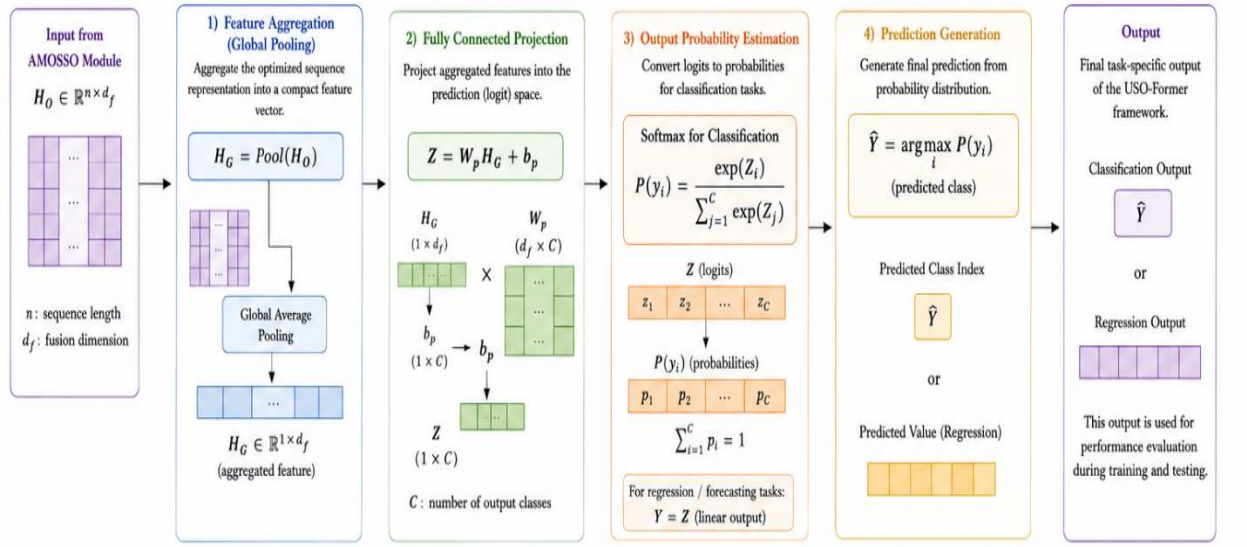


Figure 5: Prediction Layer

The input to the Prediction Layer is the optimized representation generated by AMOSSO:

$$H_O \in R^{n \times d_f} \tag{65}$$

Where (n) is the size of a sequence and (d_f) is the number of fused features. The contextual information derived using a kernelized spectral attention mechanism, preserved long-range dependencies through state-space memory evolution, and refined features via adaptive optimization comprise the sequence representation. The resulting optimized sequence representation is then aggregated to create a compact feature vector suitable for downstream learning tasks, prior to generating predictions. A global pooling operation summarizes the sequence.

$$H_G = Pool(H_O) \tag{66}$$

where $Pool\cdot$ is a global aggregation operator (the "pooled" feature representation will be indicated by (HG)). The type of pooling will depend on the application (e.g., average, max, attention) and will be selected based on the needs of that application. In this framework, global average pooling is recommended because of its low cost and stable representation. The pooled representation produced from the previous step is transformed into prediction space via a fully connected transformation:

$$Z = W_p H_G + b_p \tag{67}$$

Where (Wp) is the matrix of weights associated with the predictions; (bp) the bias vector; and (Z) is the logit output representation. The feature representation that has been extracted previously is transformed into an output representation that is specific to the task and is referenced in a space that can be used to generate predictions. For classification tasks, prediction probabilities are calculated using the SoftMax activation function:

$$P(y_i) = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)} \tag{68}$$

Here: Pyi is the probability of class i, C is the total number of output classes, and by using the SoftMax function, it guarantees that every probability will be greater than or equal to 0 and all of the probabilities from the model will equal 1 so that we can give a probability interpretation to the output of the model. When performing sequence forecasting or regression, the SoftMax function can be replaced with a linear output projection:

$$Y = Z \tag{69}$$

allowing direct prediction of continuous-valued outputs. The final prediction is obtained by selecting the most probable output class:

$$\hat{Y} = \arg\backslash\, maxP(y_i) \tag{70}$$

In the equation, the letter Y denotes the predicted output label; and for the regression task, this is directly corresponding to the predicted continuous value. The predicted value provides the final output of the proposed USO-Former architecture and thus serves as an objective measure of performance during the training and testing phases. The prediction layer will be optimized by training the network with a cross-entropy objective function for classification tasks in order to improve performance.

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \, logP(y_i) \tag{71}$$

where: (N) denotes the number of training samples, (y_i) represents the ground-truth label, $\big(P(y_i)\big)$ denotes the predicted probability. For regression tasks, Mean Squared Error (MSE) may be employed:

$$\mathcal{L}_{\mathcal{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{72}$$

To enhance predictive capabilities and generalizability throughout training, the minimisation of a given optimization goal is an objective that has to be satisfied. To convert the final feature representations from all previously discussed architectural components into task-specific outputs, the Prediction Layer will convert all of its inputs into outputs. The completed process of the end-to-end pipeline of the proposed USO-Former architecture will provide an effective vehicle for creating independent sequences on multiple tasks through the integration of feature aggregation with fully connected projection and probability estimation, resulting in the completion of the unified framework for objective-driven learning.

4. Experimental Setup

This section presents the experimental evaluation, that is conducted on a suite of long-sequence modeling benchmarks to demonstrate the utility of the USO-Former framework, measure long-range dependency capture capabilities, run benchmarks on a fixed architectural configuration under different long-context operation scenarios.

6.1. Dataset Description

The USO-Former framework has been evaluated on three widely used benchmarks for long sequence modeling. These three datasets as presented in Table 2, depict numerous long-distance contextual dependencies and are commonly applied to evaluate the proficiency of efficient attention methods, state-space methods, and long-context architectures. In addition, one of the benchmarks used in the study was the Long-Range Arena (LRA), which is a standardized benchmark designed to examine the effectiveness of sequence modeling architectures for the retention of long context information.

Table 2. Dataset Characteristics

Dataset	Domain	Task Type	Sequence Characteristics
Long Range Arena (LRA)	Long-Sequence Benchmark	Classification, Retrieval, Reasoning	Very Long Sequences
PG-19	Language Modeling	Long-Document Prediction	Book-Length Contexts
WikiText-103	Language Modeling	Text Prediction	Long Wikipedia Articles

The LRA benchmark consists of multiple tasks, including classifying long documents, sequentially retrieving long documents, processing images and solving path-finding problems. Furthermore, the sequence lengths of the tasks in the LRA benchmark far exceed the sequence lengths typically found within conventional Transformer-based benchmarks; thus, by using the benchmarks with such a diverse nature of tasks and long sequence properties, LRA offers a strong possibility of assessing the scalability and contextual learning characteristics of the proposed efficient sequence modeling frameworks.

Another benchmark dataset that was used in evaluating the USO-Former framework was the PG-19 dataset of books, which is a large number of public-domain literature in a book format. Previous language modeling datasets have typically consisted of short context sequences, however, as all book structures are retained in the PG-19 dataset; therefore, the sequences in this dataset are considerably longer. As a result of the nature of the PG-19 dataset, it is an ideal benchmark for evaluating long-range dependency learning, contextual memory retention, and understanding of sequences in relation to a long range of text. The long-document properties of the PG-19 dataset offer a great way to measure the performance of the proposed state-space memory mechanism and spectral attention mechanism.

The very next dataset that has been used for evaluation is the WikiText-103 dataset, which is a large-scale language modeling benchmark based on high-quality Wikipedia articles. Due to the high level of semantic quality and passage length found in the Wikipedia articles of the WikiText-103 dataset, effective contextual understanding will be needed for accurate prediction. Lastly, compared to smaller language modeling benchmarks, the WikiText-103 dataset offers more realistic larger document structure and the ability to evaluate both the quality of the represented information as well as the computational efficiency of the represented information will be evaluated. The WikiText-103 dataset has been widely adopted in studies that involve the examination of Transformer and state-space modeling; therefore, the WikiText-103 dataset is well-suited for evaluating performance comparison.

6.3 Evaluation Metrics

The proposed USO-Former has been evaluated using four performance metrics (Prediction Accuracy, Precision, Recall, and F1-Score). Furthermore, the computational resources required

for training and making inferences with the USO-Former have also been measured via three efficiency metrics (Memory Utilization, FLOPS, and Inference Time). As a result, there are seven evaluation metrics that can be used individually to measure the performance of the USO-Former. The first matric used for evaluation is accuracy and computed as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{73}$$

Where $TP, TN, FP, and\ FN$ denote true positives, true negatives, false positives, and false negatives, respectively. Precision evaluates the reliability of positive predictions by measuring the proportion of correctly predicted positive samples among all predicted positive samples. High precision indicates a low false-positive rate. The precision metric is defined as:

$$Precision = \frac{TP}{TP+FP} \tag{74}$$

Recall measures the ability of the model to correctly identify positive samples among all actual positive instances. This metric is particularly important when evaluating long-sequence classification tasks where missing important contextual information may negatively impact performance. Recall is computed as:

$$Recall = \frac{TP}{TP+FN} \tag{75}$$

The F1-Score provides a balanced evaluation by combining Precision and Recall into a single metric. It is particularly useful when class distributions are imbalanced. The F1-Score is calculated as:

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{76}$$

In addition to this, memory consumption is a continuous measure of the total amount of memory utilized during the training of a model and during the process of making predictions with the model. Because one of the main benefits of USO-Former is the ability to minimize the total amount of memory required for executing models that process sequences of greater than 1000 tokens, memory consumption will be an important metric to analyze when determining scalability. Memory consumption will be given in GB of RAM consumed and will include results from USO-Former as well as from the five baseline models under identical experimental conditions.

Additionally, total number of FLOP operations (floating point operations) has also been recorded. FLOPs provide a quantitative representation of the computational workload of executing forward and backward passes through a trained model. Smaller FLOP amounts indicate greater levels of computational efficiency and therefore less overhead per sequence processed. FLOP measurements are typically used to assess the relative computational workload of different deep learning architectures and to quantify the relative computational costs of those architectures in an independent way (i.e., without consideration to the specific hardware used).

6.4. Implementation and Training Configuration

To conduct a fair and reliable evaluation of the proposed framework, all experiments have been performed in the same training and deployment environment. All experiments followed a standardized methodology to remove any bias in the experimentation process and make the comparison of performance from all sequence models assessed reliable by all experimenters. The configuration settings and training specifics used to validate is summarized in Table 3.

Table 3: Implementation and Training Configuration

Parameter	Value

Deep Learning Framework	PyTorch
Optimizer	AdamW
Initial Learning Rate	0.0001
Learning Rate Scheduler	Cosine Annealing
Embedding Dimension	512
Number of Attention Heads	8
Batch Size	32
Number of Epochs	100
Dropout Rate	0.1
Sequence Lengths	1,024 – 16,384
Spectral Rank Selection	AMOSSO Optimized
AMOSSO Population Size	30
AMOSSO Iterations	100
Hardware Platform	GPU-Based Environment

7. Results and Discussion

Extensive experiments performed on various long-sequence benchmark datasets were undertaken to evaluate the utility of the proposed USO-Former. The very first experiment shows the performance of USO-Former and its comparison with other traditional models. Table 4 presents the evaluation results and indicated the highest prediction accuracy (95.18%) and F1-score (94.96%) of all algorithms. The results also suggest that the USO-Former predictively outperforms both attention-based and state-space baseline models. Furthermore, the USO-Former has the lowest memory consumption and computational workload of any algorithm evaluated, with the fastest inference time. Thus, there is a favorable tradeoff between its effectiveness to predict and efficiency to compute. In terms of prediction accuracy, the USO-Former is clearly the best performing algorithm; the baseline Transformer model provides prediction with only 91.20% accuracy, which is the lowest among the models examined in this research. Although the Transformer continues to provide strong predictive performance at moderate sequence lengths; the quadratic algorithm associated with self-attention introduces a significant computational burden, making it impractical for large-scale applications since long contexts will reduce the quality of representation. Performer implements kernel approximations of FAVOR+ to enhance algorithmic scalability over the traditional Transformer's self-attention method by decreasing its computational complexity.

Table 4: Quantitative Performance Comparison

Model	**Accuracy (%)**	**F1-Score (%)**	**Memory Usage (GB)**	**FLOPs (G)**	**Inference Time (ms)**
Transformer	91.20	90.85	12.40	210	135
Performer	92.35	92.01	8.10	145	104
Longformer	92.84	92.47	7.30	138	98
Mamba	93.76	93.42	5.20	112	79
RWKV	93.15	92.88	5.70	118	84
USO-Former	**95.18**	**94.96**	**4.60**	**96**	**71**

The kernelized attention model has improved its prediction accuracy to 92.35%, while also enhancing computational efficiency. The kernelized attention technique preserves more than 50% of the contextual representation when using the kernelized attention model as compared to other baseline algorithms. The performance of the longformer has further improved its predictive capabilities through incorporating sparse attention mechanism. The longformer algorithm has an accuracy of 92.84% and an F1-score of 92.47% when using localized and global attention patterns to process long sequences.

Another experiment determine that how well each model maintains the contextual information with respect to shifts in the length of the context being modelled. The quantitative results are presented in Table 5, while the corresponding performance trends are illustrated in Figure 6.

Table 5: Quantitative Performance Comparison

Model	1024	2048	4096	8192	16384
Transformer	91.20	89.54	87.33	84.12	79.84
Performer	92.35	91.68	90.94	89.41	87.73
Long former	92.84	92.17	91.63	90.24	88.61
Mamba	93.76	93.28	92.87	92.05	91.43
RWKV	93.15	92.76	92.13	91.32	90.54
USO-Former	95.18	94.91	94.62	94.14	93.75

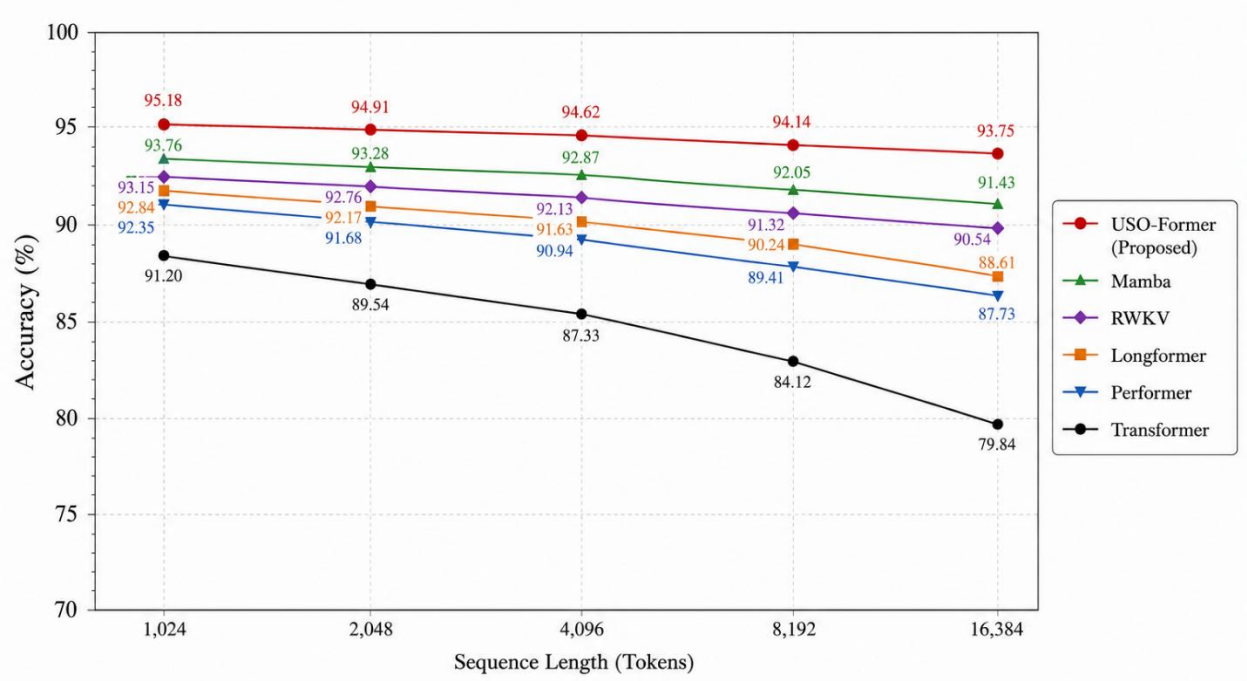


Figure 6: Accuracy Under Increasing Sequence Lengths

According to the results, all of the evaluated models experience performance degradation as the sequence length increases. The conventional transformer models show the most significant reduction in predictive capability while the proposed USO-Former remains quite stable at all sequence lengths evaluated. The transformer baseline model reaches a maximum accuracy at a sequence length of 1,024 tokens equal to 91.20%, but then degrades to 79.84% when using a sequence length equal to 16,384 tokens simultaneous. The large amount of reduction seen here can

be explained by the quadratic complexity involved with transforming self-attention into learning how contextual interaction occurs as the sequence length increases. In addition, using a large number of tokens results in a greater computer processing load and memory requirement on the server, which will have a deleterious impact upon the quality of the representation made using long-contextual sequences. In particular, the proposed USO-Former consistently reaches an accuracy of 95.18% at 1,024 tokens and 93.75% at 16,384 tokens, reflecting the least degree of performance degradation across the assessed architectures. The superior stability of the proposed framework results from the complementary use of kernelized spectral attention and spectral state-space memory modeling.

To evaluate the memory characteristics of the proposed architectural approach, the amount of memory consumed by the proposed architectural approach was measured under controlled experimental conditions, and the results were compared with the memory consumption associated with the Transformer, Performer, Longformer, Mamba and RWKV approaches. The quantitative results of the comparison are found in Table 6, and the analysis of the memory utilization trends of each of the various architectures is illustrated in Figure 7.

Table 6: Memory Consumption Comparison

Model	Memory Usage (GB)
Transformer	12.40
Performer	8.10
long former	7.30
Mamba	5.20
RWKV	5.70
USO-Former	4.60

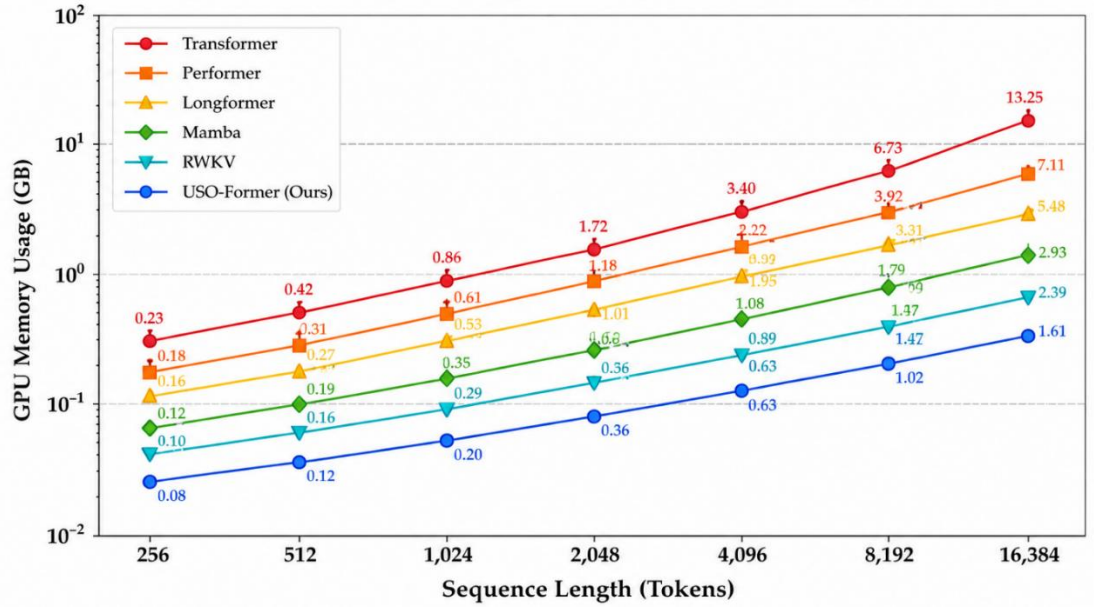


Figure 7: Memory Efficiency Comparison

The findings illustrate that the performance and efficiency of the USO-Former architecture is superior to all architectures examined in terms of memory requirements for the evaluated configuration. Specifically, memory consumption for the transformer architecture exceeded that

of USO-Former by a factor of nearly three (12.4 gigabytes). These results provide further evidence that the USO-Former architecture effectively addresses one of the major computational issues associated with the use of long-sequence learning methods. The memory growth trends also provide a better understanding of how various architectures scale as a function of sequence length. minimizes the amount of memory used while still producing strong contextual representation capabilities. These findings further support the proposed Unified Spectral Operator Framework for scalable long-sequence modeling.

Computational Efficiency is a key factor in determining the effectiveness of recent generations of sequence modelling architectures, especially for large data sets and long, continuous contexts. The results obtained for the various modelling architectures are provided in Figure 8.

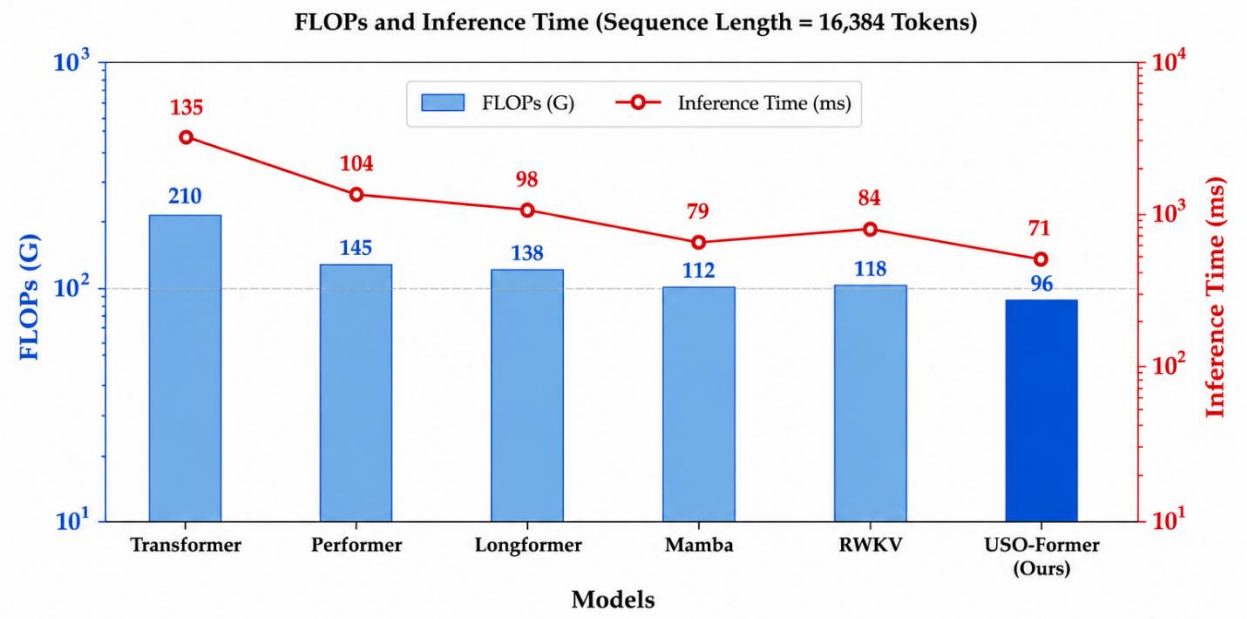


Figure 8: Computational Efficiency Comparison

The USO-Former architecture produces the best computational efficiency rating compared to all other learner architectures. In terms of computational workload, it requires 96 GFLOPs of computations; in terms of inference speed, it averages 71 milliseconds per inference. In terms of prediction accuracy, it produces the highest predictive performance. These results indicate that the USO-Former successfully overcomes many of the historical issues related to computational limitations when performing long-sequence learning

One of the main building blocks of our proposed USO-Former architecture is the Kernelized Spectral Attention Module (KSAM). The KSAM provides a method for modelling long range contextual relationships by combining kernel projection from our FAVOR+-based approach with attention as learned from an image in the spectral domain. The goal of this experiment was to ascertain how much kernelized spectral attention aids in learning contextual representations and overall predictive accuracy.

Table 7: Effect of Kernelized Spectral Attention

Configuration	Accuracy (%)	F1-Score (%)

Without Kernelized Spectral Attention	93.42	93.05
Full USO-Former	95.18	94.96

The result in Table 7 illustrates that the K-SAM significantly contributes toward improving average predictive performance. When K-SAM is removed from the model, predictive accuracy decreased from 95.18% to 93.42%, and F1-Score decreased from 94.96% to 93.05%. As such, these results illustrate the importance of K-SAM in improving built-in effectiveness of the KSA-M architecture. The extent to which K-SAM is beneficial through improved contextual learning within long-sequence modeling is further illustrated by the drop in overall model performance due to the removal of K-SAM, indicating that the model is now completely reliant on memory evolution alone.

The effect of Spectral State Space Memory Block has also been measured. Evidence from Table 8 shows that predictive performance suffers greatly when the Spectral Memory Block is removed. The accuracy of the model drops from 95.18% to 93.87% and F1-Score falls from 94.96% to 93.51%; both results confirm memory evolution's essential role in helping to conserve contextual information over long sequences as well as its significant contribution to the success of the proposed framework. The reduction in the performance of the model shows that simply having contextual attention will not enable optimal modeling of long sequences; while the Kernelized Spectral Attention Module can efficiently identify associations between sequence components, contextual interactions alone are only one way that memory blocks can help conserve long-term knowledge.

Table 8: Effect of Spectral State Space Memory Block

Configuration	Accuracy (%)	F1-Score (%)
Without Spectral State Space Memory	93.87	93.51
Full USO-Former	95.18	94.96

By having an independent mechanism to store and update latent state information continuously, the memory block allows key information to survive throughout the learning process; without this capability, the model would have diminished ability to maintain long-term contextual knowledge and, therefore, weaker predictive performance.

An ablation experiment was performed to quantify the results of using the proposed fusion strategy by replacing the adaptive gating mechanism with a simple feature concatenation process (while keeping all other aspects of the architecture the same). The aim of this experiment was to assess whether or not providing dynamic weights to both attention and memory representations offered a measurable benefit over static feature integrations.

Table 9: Effect of Adaptive Fusion Mechanism

Configuration	Accuracy (%)	F1-Score (%)
Without Adaptive Fusion	94.01	93.68
Full USO-Former	95.18	94.96

As presented in Table 9, model accuracy was decreased from 95.18% (with the Adaptive Fusion Mechanism) to 94.01% (with a simple integration strategy) as well as F1-Score, from 94.96% (with the Adaptive Fusion Mechanism) to 93.68% (with a simple integration strategy), produced a reduction in performance. While performance reduction of the other proposed attention and memory modules demonstrated greater variability than that of the adaptive representation system, it is clear that formulation of the Adaptive Fusion Method is an advantage to the overall system performance of the model. The reason for the performance decrease when using a static feature integration method was due to static integration methods being unable to account for different characteristics of a particular input sequence of varying lengths.

To evaluate the success of AMOSSO, an ablation experiment was performed by removing AMOSSO from the method and fixing all architecture parameters at the start of the training. A performance comparison between fixed and dynamic parameterized configurations is shown in Table 10.

Table 10: Effect of AMOSSO Optimization Strategy

Configuration	**Accuracy (%)**	**F1-Score (%)**
Without AMOSSO Optimization	94.26	93.94
Full USO-Former	**95.18**	**94.96**

Table 10 shows that removing the adaptive multi-objective spectral swarm optimization (AMOSSO) optimization strategy reduces predictive accuracy from 95.18% to 94.26% and the F1-Score from 94.96% to 93.94%. The testing showed the same results as data in the same category for both model performance and training efficiency. Although the decline in accuracy due to AMOSSO is less than the attention and memory component errors, they showed the need for and support of optimizing to find the best overall solution to this architecture.

Finally, an ablation study was conducted to show the contribution of each module to predicted performance. The quantitative results of this analysis are summarized in Table 11, while the corresponding performance comparison is illustrated in Figure 9.

Table 11: Overall Ablation Analysis

Configuration	**Accuracy**
Full USO-Former	95.18
Without Kernelized Spectral Attention	93.42
Without Spectral State Space Memory	93.87
Without Adaptive Fusion	94.01
Without AMOSSO Optimization	94.26

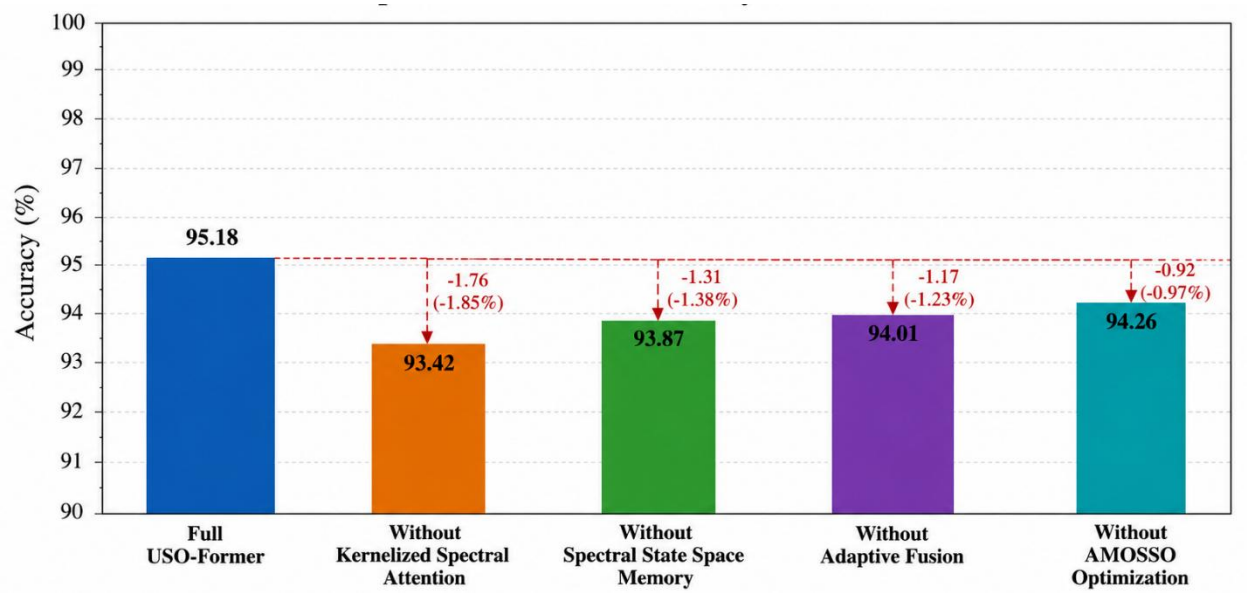


Figure 9: Component Contribution Analysis of USO-Former

As demonstrated, if any individual component is removed from the architecture there will be a measurable drop in the prediction accuracy. Thus, it can be inferred that every module provides a beneficial contribution towards the overall performance of the framework. Also the findings show that the improved performance of the proposed architecture was not specifically due to any one particular learning method but rather the combined synergistic interaction among five learning methods, which include attention-based contextual learning, spectral representation modeling, state-space evolution of memory, adaptive feature fusion, and multi-objective optimization. The greatest loss in predictive accuracy occurs when the Kernelized Spectral Attention Module is disabled. When this component was removed from the architecture predictive accuracy decreased from 95.18% down to 93.42%, which indicates a loss of 1.76 percentage points (or approximately 1.76%) in predictive performance.

9. Conclusion and Future Work

We propose the USO-Former that combines spectral state-space memory modelling for hierarchical linguistic contextual learning. With the help of local token attention, segment-level semantic representation, global contextual interaction and recurrent memory evolution, we propose an efficient framework for long textual sequence modelling. The proposed USO-Former architecture includes five unique architectural components: multi-scale embedding layer, kernelized spectral attention module, spectral state space memory block, adaptive fusion mechanism, and an adaptive multi-objective spectral swarm optimization (AMOSSO) strategy that are effective at providing robust representation learning, modeling of long-range dependencies and optimizing the parameterization of models for application to long-sequence learning problems. The USO-Former was tested using three long-range datasets (i.e. Long Range Arena (LRA), PG-19, and WikiText-103) in which the performance of the USO-Former was compared against other representative sequence modeling architectures (i.e., Transformer, Performer, longformer, Mamba, RWKV). The results showed the USO-Former has superior predictive performance while consuming less memory and computational resources than the other five models. Specifically, the

architecture of the USO-Former had the highest accuracy and F1 Score of all of the architectures tested while using the least amount of memory and performing the fewest number of computations as compared to the other models. Future work can investigate the incorporation of dynamic spectral operators that adapt how frequency domain representations are modified in real time to reflect the characteristics of the sequences being modeled at any moment. Building upon this concept, further evolving the research may also investigate the use of dynamic spectral learning mechanisms to eventually enhance contextual understanding and flexibility of representation across multiple use cases.

Declarations

Funding: The author received no financial support for the research, authorship, or publication of this work.

Author Contributions: Jinhang Xu proposed the research idea, designed the research plan, managed the project, and provided experimental equipment, materials, and samples.

Consent to Publish declaration: not applicable.

Ethics and Consent to Participate declarations: not applicable.

References

1. Zhu, J., Huang, S., Wang, P., Xu, Z., Liu, J., Wang, B., ... & Wang, L. (2026). A review on large language models for industrial embodied intelligence. *Advanced Engineering Informatics*, *73*, 104602.
2. Zhang, Z., Pei, H., Guo, J., Wang, T., Feng, Y., Sun, H., ... & Sun, A. (2026, April). Onetrans: Unified feature interaction and sequence modeling with one transformer in industrial recommender. In *Proceedings of the ACM Web Conference 2026* (pp. 8162-8170).
3. Ghojogh, B., & Ghodsi, A. (2026). Attention Mechanism and Transformers. In *Elements of Deep Learning* (pp. 231-257). Cham: Springer Nature Switzerland.
4. Machemedze, J., & Ndlovu, B. (2026). Transformer-Based Models for Electronic Health Records and Omics in Healthcare: A Systematic Literature Review. *Journal of Applied Informatics and Computing*, *10*(1), 90-105.
5. Wu, H., Guo, M., Ma, Y., Sun, Y., Wang, J., Matusik, W., & Long, M. (2026). Flashbias: Fast computation of attention with bias. *Advances in Neural Information Processing Systems*, *38*, 20729-20756.
6. Chauhan, A., Chaudhary, A., & Jain, S. (2026). Enhanced Spatial–Temporal Transformer Network for Robust Micro-Expression Temporal Localization and Recognition in Long Video Sequences. *Concurrency and Computation: Practice and Experience*, *38*(10), e70752.
7. Zhou, R., Jia, Q., Chen, B., Xu, P., Sun, Y., Lou, S., ... & Chen, E. (2026). A Survey of User Lifelong Behavior Modeling: Perspectives on Efficiency and Effectiveness.
8. Wang, Y., Wu, H., Dong, J., Liu, Y., Wang, C., Long, M., & Wang, J. (2026). Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

9. Khanmohammadi, R., Miahi, E., Kaur, S., Smiley, C., Brugere, I., Thind, K. S., & Ghassemi, M. M. (2026, March). How reliable are confidence estimators for large reasoning models? a systematic benchmark on high-stakes domains. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1669-1754).
10. Zhang, J., Su, R., Liu, C., Wei, J., Wang, Z., Wang, H., ... & Fu, T. Efficient Attention Methods: Hardware-efficient, Sparse, Compact, and Linear Attention.
11. Zhang, Y., Sun, W., Sun, X., Ding, W., Xie, R., Wang, H., ... & Wang, Y. (2026). A Survey of Linear Attention: Algorithm, Theory, Application, and Infrastructure. *Authorea Preprints*.
12. Wang, Y., Wu, H., Dong, J., Liu, Y., Wang, C., Long, M., & Wang, J. (2026). Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
13. Nour, M., Şentürk, Ü., Akgül, A., & Polat, K. (2026). Quantitative EEG-based autism spectrum disorder detection using neural sequence models. *Computers and Electrical Engineering*, *131*, 110962.
14. Mubashira, M., Dilshah, F., Fathima, A., Arafath, Y., Hassan, S., Kiran, G. S., & Selvin, J. Insights into prebiotics, probiotics, and synbiotics towards growth, performance, immune responses, and disease resistance in aquaculture. In *Innovative Solutions and Strategies for Fish Health Management* (pp. 328-340). CRC Press.
15. Zheng, Y., Zhang, X., Zhou, Y., Zhang, Y., Zhang, T., & Farmani, R. (2025). Deep representation learning enables cross-basin water quality prediction under data-scarce conditions. *NPJ Clean Water*, *8*(1), 33.
16. Athanasopoulou, K., Michalopoulou, V. I., Scorilas, A., & Adamopoulos, P. G. (2025). Integrating artificial intelligence in next-generation sequencing: advances, challenges, and future directions. *Current Issues in Molecular Biology*, *47*(6), 470.
17. Osofuye, O. D. (2025). An Adaptive Multi-Objective Memetic Algorithm (AMOMA) for the Hyperparameter Tuning of LightGBM.
18. Yang, J., et al. (2025). Linear Attention for Efficient Bidirectional Sequence Modeling. arXiv.
19. An, Z., et al. (2025). State-Space Model Meets Linear Attention: A Hybrid Framework for Efficient Sequence Modeling. Remote Sensing, 17(17), 2969.
20. Chen, W., Liu, L., & Gao, R. (2026). Spatial-Spectral Enhanced Mamba with Diffusion Prior for Spectral Super-Resolution. Journal of King Saud University – Computer and Information Sciences, 38, Article 84.
21. Sellam, A. Z., Zidi, F. A., Bekhouche, S. E., Houhou, I., Tliba, M., Distante, C., & Hadid, A. (2026). VP-Hype: A Hybrid Mamba-Transformer Framework with Visual-Textual Prompting for Hyperspectral Image Classification. arXiv.
22. Djuhera, A., Gacanin, H., & Boche, H. (2026). MambaCSP: Hybrid-Attention State Space Models for Hardware-Efficient Channel State Prediction. arXiv.
23. Nguyen, T. D., Bae, S., Chung, J. S., & Kim, J.-H. (2026). MamTra: A Hybrid Mamba-Transformer Backbone for Speech Synthesis. arXiv.
24. Alkhatib, M. Q. (2026). ConvVitMamba: Efficient Multiscale Convolution, Transformer, and Mamba-Based Sequence Modelling for Hyperspectral Image Classification. arXiv.
25. Zhan, Z., Zhao, J., Zhu, Z., & Tang, J. (2025). Overcoming Long-Context Limitations of State-Space Models via Context-Dependent Sparse Attention. arXiv.

26. Halloran, J. T., Gulati, M., & Roysdon, P. F. (2025). Mamba State-Space Models Are Lyapunov-Stable Learners. arXiv.
27. Alsmann, E., & Lange, M. (2025). Characterizing State Space Model and Hybrid Language Model Efficiency for Long-Context Inference. arXiv.
28. Tu, W., Li, J., Xiao, F., Wang, X., & Lu, Y. (2026). Integrating Large Language Models into Traffic Systems: Integration Levels, Capability Boundaries, and an Information-Theoretic Perspective. *Entropy*, *28*(2), 211.
29. Zhang, H., Huang, S. L., KURUOGLU, E., & Wang, Y. (2026). Multi-Kernel Correlation-Attention Vision Transformer for Enhanced Contextual Understanding and Multi-Scale Integration. *Advances in Neural Information Processing Systems*, *38*, 70338-70364.
30. Chai, Z., Ren, Q., Xiao, X., Yang, H., Han, B., Zhang, S., ... & Wu, D. (2025, September). Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems* (pp. 247-256).