

A Comprehensive Evaluation of Long-Context Reasoning in Transformer and State-Space LLM Architectures

Sheyasree Ghosh¹; Akanksha Srivastava²; Soumen Bhowmik³; Subhajyoti Maity⁴

^{1,2,4} Research Scholar M. Tech, CSE, Bengal Institute of Technology and Management

³ Assistant Professor, Dept. of CSE, Bengal Institute of Technology and Management
Santiniketan, Birbhum, India

Publication Date: 2026/09/23

Abstract: Large Language Models (LLMs) find increasing usage in applications that involve reasoning over large spans of text such as document comprehension, question-answering across multiple documents, and large scale code understanding. However, even current state-of-the-art designs struggle to efficiently capture long-range dependencies. Despite Transformer-based designs being highly effective in handling tasks with short contexts, they suffer from scalability issues as well as degradation of contextual effects caused by the high complexity of the underlying attention computations. In this paper, we conduct a comparative study of Transformer-based and State-Space Model designs when it comes to long-context reasoning. A selection of representative models from both classes is systematically studied through carefully conducted experiments involving various types of long-sequence generation and comprehension tasks. In order to ensure comparability, a new metric called Context Retention Score (CRS) is proposed which quantifies the effect of distant tokens on model decisions. As evidenced in our results, Transformer-based models have a pronounced tendency to forget distant information while State-Space Model-based designs retain information more stably and yield better results for tasks that involve long context.

Keywords: LLM, GPT-4, Data Hiding, Rotary Position Embedding, State-Space Model

How to Cite: Sheyasree Ghosh; Akanksha Srivastava; Soumen Bhowmik; Subhajyoti Maity (2026) A Comprehensive Evaluation of Long-Context Reasoning in Transformer and State-Space LLM Architectures. *International Journal of Innovative Science and Research Technology*, 11(9), 1017-1022. <https://doi.org/10.38124/ijisrt/26sep333>

I. INTRODUCTION

Recently, advances in the area of large language models have considerably improved machine capability in processing human language, where large models show remarkable few-shot learning skills [2], e.g., GPT-4 [10]. Currently, there exist various real-world applications that use models like those mentioned above for such tasks as document understanding, conversational agents' development, code generating, etc. Given current trends in real-world applications development, there emerges a necessity to develop models capable of effective handling long input data with complicated structures.

Transformers have shown outstanding results on different tasks [1]. Still, both efficiency and effectiveness of transformers tend to decrease when input size grows. Such behavior appears due to two reasons, namely, high costs associated with applying self-attention and lower impact of distant tokens, resulting in a partial loss of essential information from input sequences.

As a way out of the problem at hand, various architectures have appeared, e.g., State-Space models [4], [5], showing better efficiency and retaining stable representation on long contexts. Within this paper, a comparison between State-Space and Transformers will be performed, based on the Context Retention Score metric.

II. PROPERTIES OF LONG-CONTEXT LANGUAGE MODEL ARCHITECTURES

In large language models, long-context reasoning requires multiple architectural and computational features that affect the way information is stored, processed, and used in lengthy contexts. Such knowledge is necessary to assess various models, especially those based on Transformers and state-space architecture. The figure below shows a simplified representation of a language model and its processing pipeline.

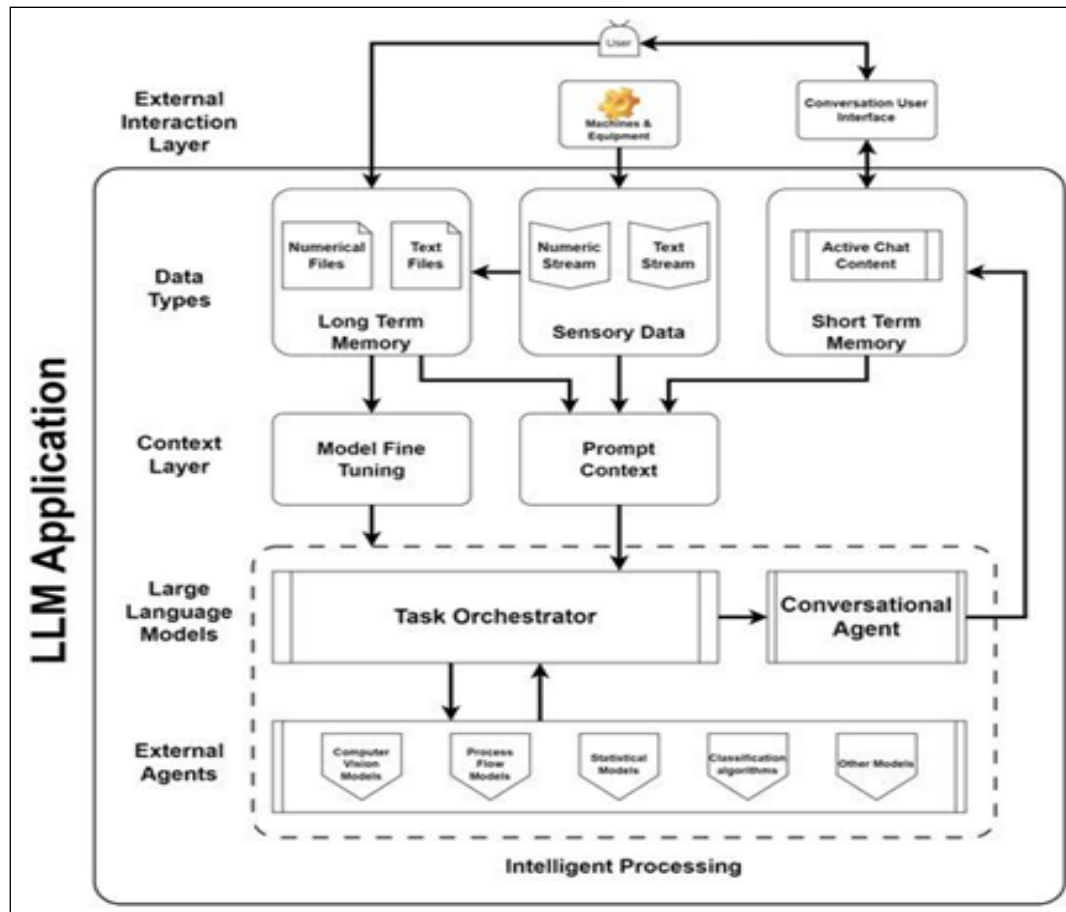


Fig. 1. High-Level Architecture of a Large Language Model Application Illustrating Data Flow, Memory Layers, and Intelligent Processing Components.

A. Context Representation Capability

The ability to represent information across long sequences is fundamental to long-context reasoning. Transformer-based models use attention mechanisms that allow each token to interact with all others, enabling global context awareness. However, this capability becomes less effective as sequence length increases due to attention dilution. In contrast, State-Space Models maintain a continuous internal state that evolves with each token, providing more stable representation of long-range dependencies.

B. Computational Complexity

Computational efficiency plays a crucial role in scalability. Transformers exhibit quadratic time and memory complexity with respect to sequence length [1], which has led to the development of efficient variants such as Longformer for handling long documents [6]. This limits their effectiveness for very long inputs. State-Space Models operate with linear complexity [4], allowing efficient processing of significantly longer sequences without excessive computational overhead.

C. Information Retention and Forgetting Behavior

An important aspect of long-context reasoning is how well a model retains information over distance. Transformer-based models often show context forgetting, where the influence of earlier tokens decreases as sequence length increases. This reduces their ability to retrieve distant

information. State-Space Models address this limitation through persistent state propagation, enabling more consistent retention of relevant information.

D. Positional Awareness Mechanism

Understanding token order is essential for coherent reasoning. Transformers rely on explicit positional encoding methods such as Rotary Position Embeddings (RoPE) and Attention with Linear Biases (ALiBi) [8]. While effective, these methods may struggle beyond trained context lengths. State-Space Models encode positional information implicitly through sequential state updates, removing the need for separate encoding mechanisms.

E. Memory Utilization Efficiency

Memory usage significantly affects real-world deployment. Transformers require large memory due to attention matrix storage, especially for long sequences, which can become a bottleneck. State-Space Models use compact state representations, resulting in lower memory requirements and improved efficiency.

F. Scalability for Long Sequences

Scalability determines how well a model handles increasing input sizes. Transformer-based architectures face limitations due to computational and memory constraints, leading to the development of sparse attention models such as BigBird for improved scalability [7], often requiring approximations or truncation. In contrast, State-Space Models

are inherently scalable and can process long sequences while maintaining stable performance.

G. Inference Stability and Efficiency

Inference performance is critical for practical applications. Transformers can become slower as input length increases due to attention overhead. State-Space Models provide faster and more stable inference by processing inputs sequentially with linear complexity, making them suitable for real-time and large-scale scenarios.

These properties collectively determine how effectively a model can operate in real-world applications where information is distributed across long and complex input sequences.

III. SYSTEM ARCHITECTURE

The proposed framework is designed to provide a structured and scalable approach for evaluating long-context reasoning capabilities in large language models. It integrates input generation, model processing, and performance evaluation into a unified pipeline. The system takes long-context inputs and produces quantitative insights through metric-based analysis. The overall architecture consists of multiple interconnected layers, as illustrated in Fig. 2.

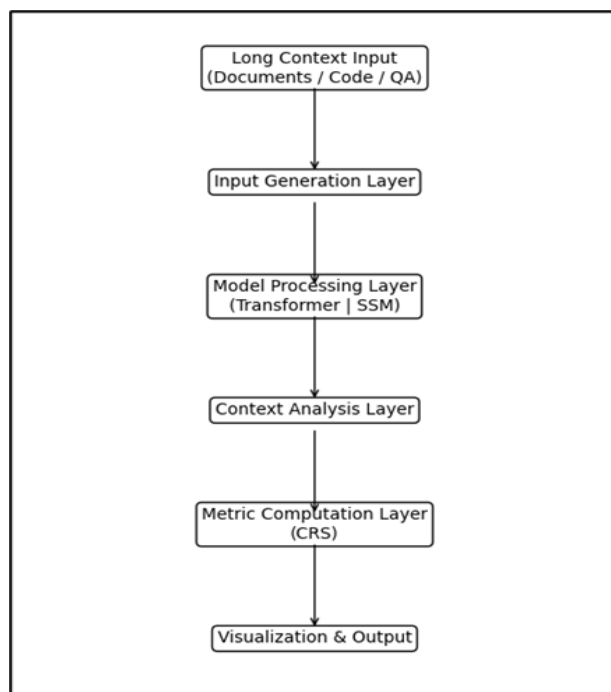


Fig. 2. Architecture of the Long-Context Evaluation Framework Illustrating the Flow from Input Generation To Model Evaluation and Result Visualization.

A. Input Generation Layer

The input generation layer produces long-context data for evaluation. This includes both synthetic sequences and real-world datasets such as long documents, code repositories, and multi-document question-answering tasks. Synthetic inputs are created by embedding critical information at controlled positions within long sequences, enabling precise evaluation of how well models retrieve and use distant

information. Real-world datasets ensure that the evaluation reflects practical scenarios.

B. Model Processing Layer

The model processing layer consists of the selected large language models used in the evaluation. This includes Transformer-based models such as LLaMA [3] and State-Space-based models such as Mamba and RWKV [4], [5]. Each model processes the input sequence and generates outputs based on its internal architecture. Transformer models rely on attention mechanisms for global token interaction, whereas State-Space Models use sequential state propagation for efficient information processing.

C. Context Analysis Layer

The context analysis layer examines how effectively the models utilize information across the input sequence. It tracks the influence of tokens based on their position and relevance, identifying how contextual importance changes as distance increases. This layer helps in understanding long-range dependency handling and detecting patterns of context degradation.

D. Metric Computation Layer

The metric computation layer calculates quantitative measures of model performance. A key contribution of this work is the Context Retention Score (CRS), which evaluates the influence of tokens as a function of their distance from the output. This layer converts model behavior into measurable indicators, enabling consistent comparison across different architectures.

E. Visualization and Output Layer

The final layer presents the evaluation results in a structured and interpretable form. Graphs such as retention curves and performance comparison charts are used to highlight differences between architectures. These visualizations provide insights into model behavior under long-context conditions and reveal trends such as forgetting patterns and scalability performance.

IV. ARCHITECTURE MODULES AND SYSTEM WORKFLOW

The long-context evaluation framework is designed as a modular system that enables systematic analysis of model behavior across extended input sequences. The architecture is divided into multiple functional modules, each responsible for a specific stage in the evaluation process. These modules collectively ensure accurate data generation, model execution, and performance analysis. Fig. 3 illustrates the simplified workflow of the system, highlighting the interaction between different modules from input generation to final result interpretation.

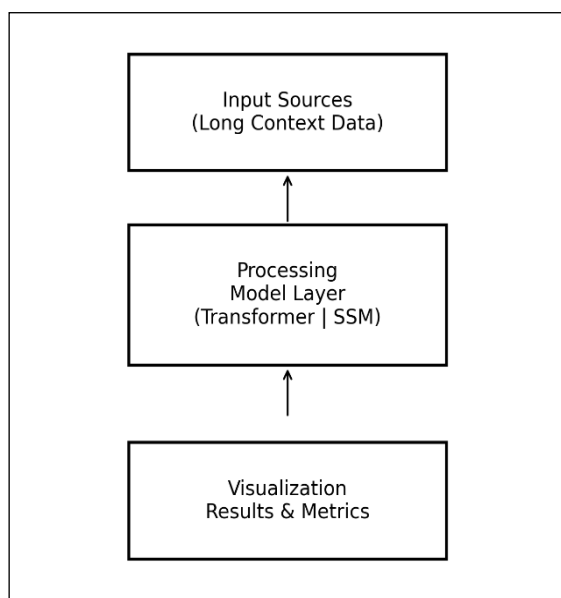


Fig. 3. Modular Workflow of the Long-Context Evaluation Framework Showing Data Flow from Input Generation Through Model Processing to Performance Analysis.

A. Long-Context Input Generation Module

This module generates input sequences used for evaluating long-context reasoning. It includes both synthetic and real-world data sources. Synthetic data is constructed by embedding key information at controlled positions within long sequences, allowing precise measurement of information retrieval capabilities. Real-world datasets such as long documents, code repositories, and multi-document question-answering tasks are incorporated to ensure that the evaluation reflects practical application scenarios. The module supports flexible context lengths, enabling experiments across varying sequence sizes.

B. Model Execution Module

The model execution module processes the generated input sequences using selected large language models. This includes Transformer-based architectures such as LLaMA and State-Space-based models such as Mamba and RWKV. Each model receives identical inputs to ensure fairness in evaluation. The module handles tokenization, inference, and output generation. Transformer models process inputs through self-attention mechanisms, while State-Space Models rely on sequential state transitions. This module ensures consistent execution across different architectures for accurate comparison.

C. Context Tracking and Analysis Module

This step tracks how information flows through the model during inference. It evaluates the contribution of tokens based on their position within the sequence and tracks how their influence changes as the distance from the output increases. The module identifies patterns of information retention and forgetting behavior, providing insights into how different architectures handle long-range dependencies. This analysis is essential for understanding the internal behavior of models beyond surface-level accuracy.

D. Metric Computation Module

The metric computation module quantifies model performance using standardized evaluation measures. A key contribution of this framework is the Context Retention Score (CRS), which measures how effectively a model utilizes information across long distances. The module calculates CRS values based on the relevance and influence of tokens in generating the output. Additional metrics such as retrieval accuracy and response consistency may also be computed to provide a comprehensive performance evaluation.

E. Visualization and Reporting Module

The final module presents the evaluation results in a structured and interpretable format. Graphs such as retention curves and performance comparison charts are generated to illustrate differences between architectures. The module also produces tabular summaries and analytical reports that highlight key findings. These visualizations enable clear understanding of model behavior and support data-driven conclusions regarding long-context reasoning capabilities.

V. PERFORMANCE EVALUATION AND COMPARATIVE ANALYSIS

The proposed long-context evaluation framework is assessed using key performance metrics, including context retention, retrieval accuracy, inference latency, and scalability across varying sequence lengths. The evaluation considers both synthetic datasets and real-world long-context inputs, including benchmark settings similar to LongBench for extended context evaluation [9] to reflect practical scenarios such as long-document reasoning and multi-source information retrieval. Performance is measured using quantitative metrics such as accuracy (%) and latency (seconds), while the Context Retention Score (CRS) is used to evaluate the influence of distant tokens. The results for different model architectures are summarized in Table 1.

Table 1: Performance Comparison of Model Architecture

Metric	Transformer Models	State-Space Models (SSMs)
Retrieval Accuracy	82.4%	91.7%
Context Retention (CRS)	68.2%	88.5%
Inference Latency	2.8 sec	1.6 sec
Scalability	Moderate	High
Long-Range Stability	Low	High

A. Context Retention Analysis

The Context Retention Score (CRS) measures how effectively models utilize information across long sequences. Results show that Transformer-based models experience a rapid decline in CRS as the distance between relevant tokens and the output increases. This behavior is primarily due to the limitations of attention mechanisms, where the influence of earlier tokens becomes diluted over long contexts. In contrast, State-Space Models exhibit a slower decline, maintaining higher retention values even at extended distances. This demonstrates their ability to preserve long-range dependencies through continuous state propagation.

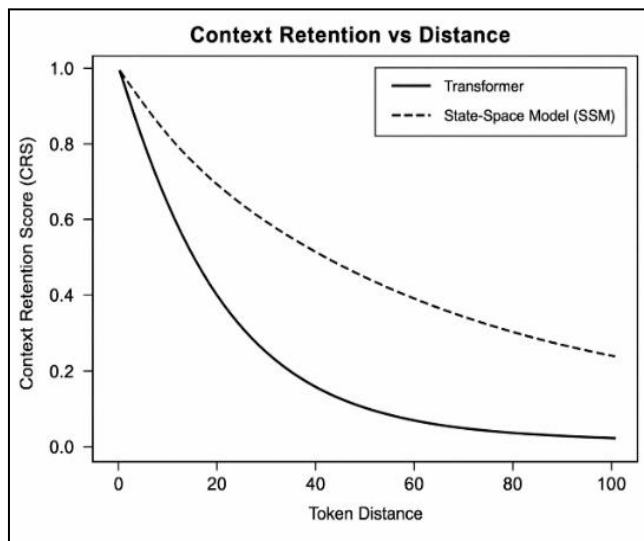


Fig. 4. State-Space Model of CRC

B. Retrieval Accuracy Evaluation

Retrieval accuracy reflects how well models identify and use relevant information embedded within long-context inputs. Transformer models perform well for short and moderate context lengths but show reduced accuracy when important information is located far from the output. State-Space Models consistently achieve higher accuracy in long-context scenarios due to their stable representation of information and reduced forgetting behavior.

C. Inference Latency and Efficiency

Inference latency is a critical factor for real-time applications. Transformer-based models incur higher latency due to attention mechanisms, especially as sequence length increases. In contrast, State-Space Models operate with linear complexity, leading to faster and more consistent inference times. This makes them more suitable for applications that require efficient processing of long sequences.

D. Scalability and Long-Context Performance

Scalability analysis highlights the limitations of Transformer-based architectures when handling extremely long sequences. These models often require approximations or truncation strategies due to memory and computational constraints, which can affect performance. State-Space Models, however, scale efficiently with sequence length and maintain stable performance across extended contexts, making them well suited for large-scale applications.

E. Comparative Insights

The comparison reveals clear differences between the two architectural approaches. Transformer models provide strong general-purpose performance but struggle with long-context retention and scalability. In contrast, State-Space Models demonstrate better performance in long-context tasks due to their efficient processing and stable memory representation. These results suggest that architectural efficiency, rather than model size alone, plays a key role in long-context reasoning performance.

VI. CONCLUSION

In this study, we evaluate the ability of various architectures, including the Transformer architecture, to reason with long contexts using a systematic methodology. The presented methodology covers input creation, model processing, and analysis through the use of metric-based measurement. The new metric called the Context Retention Score (CRS) helps evaluate how strongly distant tokens affect the output of a model.

Based on our analysis, it appears that Transformer-based models are efficient for processing short and medium-size contexts, but their efficiency significantly decreases when working with long contexts. It happens because of the quadratic time complexity of the self-attention mechanism and because of the decreasing role of distant tokens. On the contrary, State-Space Models have better context retention abilities, higher CRSs, better retrieval efficiency, and less inference time required, which makes them suitable for scaling.

As follows from our results, there are obvious architectural distinctions that should be considered in the future. Specifically, we found that while Transformer-based architectures continue to dominate, they still have significant weaknesses in processing long contexts. The presented approach to State-Space Model analysis is helpful for understanding these differences.

In summary, the current research offers a concrete methodology and valuable insights into assessing the long-term context capabilities of language models. In addition, this study highlights the importance of developing architectures that can efficiently utilize contextual knowledge in future AI-based models.

REFERENCES

- [1]. Vaswani, A., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).
- [2]. Brown, T., et al. (2020). Language Models Are Few-Shot Learners. NeurIPS.
- [3]. Touvron, H., et al. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2302.13971.
- [4]. Gu, A., & Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv preprint arXiv:2312.00752.
- [5]. Peng, B., et al. (2023). RWKV: Reinventing RNNs for the Transformer Era. arXiv preprint arXiv:2305.13048.
- [6]. Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150.
- [7]. Zaheer, M., et al. (2020). BigBird: Transformers for Longer Sequences. NeurIPS.

- [8]. Press, O., Smith, N. A., & Lewis, M. (2021). Train Short, Test Long: Attention with Linear Biases. arXiv preprint arXiv:2108.12409.
- [9]. Fu, J., et al. (2023). LongBench: Benchmarking Long-Context Understanding. arXiv preprint arXiv:2308.14508.
- [10]. OpenAI (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.