myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

Google's Neural Memory Architecture: A Comprehensive Review of the Titans Framework

1st Piyush Junghare
CSE(Cyber Security)
GHRCEM
Nagpur, India
piyushjungharecyb@ghrietn.raisoni.net

2nd Jay Chheniya
CSE(Cyber Security)
GHRCEM,
Nagpur, India
jaychheniyacyb@ghrietn.raisoni.net

3rd Mohit Behare
CSE(Cyber Security)
GHRCEM
Nagpur, India
mohitbeharecyb@ghrietn.raisoni.net

4th Pratik Kashte
CSE(Cyber Security)
GHRCEM,
Nagpur, India
pratikkashtecyb@ghrietn.raisoni.net

5th Saniya Belekar
CSE(Cyber Security)
GHRCEM
Nagpur, India
saniya.belekar.cyb@ghrietn.raisoni.net

6th Vaishnavi Dhoble
CSE(Cyber Security)
GHRCEM,
Nagpur, India
vaishnavi.dhoble.cyb@ghrietn.raisoni.net

7th Shalini Kumari
CSE(Cyber Security)
GHRCEM,
Nagpur, India
shalini.kumari@ghrau.edu.in

Abstract

This paper presents a comprehensive review of Google's Titans architecture, a groundbreaking neural memory framework designed to address the limitations of traditional Transformer models in handling long-term context. Titans introduces a novel neural long-term memory module that learns to memorize at test time, enabling language models to retain, update, and recall information over millions of tokens without the quadratic computational cost associated with standard attention mechanisms. The architecture combines three interconnected memory systems: short-term attention-based memory, a deep neural long-term memory module, and persistent memory for task-specific knowledge. This review examines the theoretical foundations, architectural components, and design principles that enable Titans to achieve superior performance on long-context tasks while maintaining computational

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

efficiency. The paper provides detailed analysis of the three architectural variants Memory as Context (MAC), Memory as Gate (MAG), and Memory as Layer (MAL) and discusses their respective strengths and trade-offs in different application scenarios.

Keywords: Deep learning, long-term memory, neural architecture, sequence modelling , test-time learning, Transformers

INTRODUCTION

The Transformer architecture, introduced by Vaswani et al. in 2017, revolutionized natural language processing by enabling parallel processing of sequences through the attention mechanism [2]. However, despite their remarkable success, Transformers face a fundamental limitation: their attention mechanism scales quadratically with sequence length, effectively constraining the context window to a fixed length. This limitation has become increasingly problematic as applications demand processing of longer documents, extended conversations, and large-scale sequential data.

Over the past decade, researchers have explored various approaches to address this challenge. Recurrent neural networks (RNNs) and their variants, including LSTMs and GRUs, compress information into fixed-size hidden states but struggle with long-term dependencies due to vanishing gradients [5]. More recently, linear recurrent models and state space models (SSMs) such as Mamba have offered linear complexity but at the cost of reduced expressiveness in capturing complex dependencies [3][4]. The RWKV architecture presents another promising direction, combining efficient parallelizable training of transformers with efficient inference of RNNs [11].

In December 2024, Google Research introduced Titans, a new family of architectures that fundamentally reimagines how neural networks handle memory [1]. The key innovation lies in treating memory not as a passive storage mechanism but as an active learning system. Titans' neural long-term memory module learns to memorize at test time, using gradient-based optimization to selectively store surprising or important information while maintaining fast parallelizable training and efficient inference. This approach draws inspiration from meta-learning principles and fast weight networks [6][13].

The significance of Titans extends beyond its technical innovations. By enabling language models to process millions of tokens with near-constant computational complexity, Titans opens new possibilities for applications requiring extensive context understanding, from analyzing entire codebases to processing lengthy legal documents and conducting comprehensive literature reviews. This review paper provides a comprehensive analysis of the Titans framework, examining its theoretical foundations, architectural components, and potential implications for the future of deep learning.

THEORETICAL FOUNDATIONS OF NEURAL MEMORY

Memory Systems in Neural Networks

The design of Titans is inspired by human cognitive memory systems, which comprise distinct but interconnected components. Human memory is broadly categorized into short-term (working) memory and long-term memory. Short-term memory has limited capacity but provides precise, immediate access to recent information. Long-term memory, conversely, has vast capacity but requires mechanisms for encoding, consolidation, and retrieval [7][8].

In the context of neural architectures, attention mechanisms serve as an analog to short-term memory, providing accurate modeling of dependencies within a limited context window. However, the quadratic

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

complexity of attention restricts its effective range. Titans addresses this by introducing a dedicated neural long-term memory module that complements attention, creating a hybrid architecture that combines the precision of short-term attention with the capacity of long-term neural memory. Memory-augmented neural networks have shown promise in enhancing the capabilities of large language models by storing attention keys and values in external memory [12].

The Surprise Metric

A central concept in Titans' memory system is the "surprise metric," which determines what information should be retained in long-term memory. Drawing from human psychology, where unexpected events are more memorable than routine occurrences, Titans uses the gradient of the neural network with respect to the input as a measure of surprise. When the model encounters input that significantly deviates from its expectations (high gradient), the surprise metric signals that this information should be prioritized for long-term storage [1].

Mathematically, the surprise for an input token is computed as the gradient of the associative memory loss with respect to that token. This gradient-based approach provides a principled mechanism for selective memorization, allowing the model to focus its limited memory capacity on the most informative and unexpected inputs while filtering out redundant or predictable information. The surprise metric incorporates both momentary surprise (measured by the gradient) and past surprise (a decaying memory of recent surprising events), creating a comprehensive measure of information novelty [1][9].

Meta-Learning and Test-Time Adaptation

Titans employs a meta-learning framework where the neural memory module learns in an inner loop while the overall architecture is trained in an outer loop. This approach is related to test-time training (TTT) methods, where models adapt their parameters at inference using self-supervised learning from unlabeled data to mitigate domain shifts [14]. Rapid Network Adaptation (RNA) represents another approach to test-time adaptation, using a learning-based function as an amortized optimizer for efficient network adaptation [15].

The memory update mechanism in Titans employs an associative memory loss that learns mappings between keys and values. Given an input token, the module computes a key-value pair and updates its weights to minimize the reconstruction error. This optimization occurs in the inner loop of a meta-learning framework, where the memory module's weights are updated while the key and value projection matrices remain fixed as hyperparameters [1][13].

CORE ARCHITECTURAL COMPONENTS

Short-Term Memory Module

The short-term memory module in Titans is implemented using attention mechanisms, specifically sliding window attention (SWA). This design choice limits the attention scope to a fixed-size local window, reducing computational complexity from quadratic to linear while maintaining the ability to model precise dependencies within the window. The sliding window approach ensures that each token attends only to its immediate context, typically ranging from a few hundred to a few thousand tokens depending on the configuration [1].

The short-term memory module serves as the "core" processing unit of the Titans architecture, handling the primary flow of information and making decisions about what should be stored in long-term memory. By limiting the attention window, this module ensures computational efficiency while still capturing the fine-grained dependencies necessary for accurate sequence modelling . This approach addresses the fundamental limitation of standard Transformers while preserving their expressive power for local context [2][10].

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

Neural Long-Term Memory Module

The neural long-term memory module is the cornerstone innovation of Titans. Unlike traditional recurrent models that compress information into fixed-size vectors or matrices, this module is implemented as a deep neural network specifically, a multi-layer perceptron (MLP) that learns to memorize historical context. The module operates as a meta-learner, optimizing its parameters at test time to store and retrieve information efficiently [1][6].

The memory update mechanism employs an associative memory loss that learns mappings between keys and values. Given an input token, the module computes a key-value pair and updates its weights to minimize the reconstruction error. This optimization occurs in the inner loop of a meta-learning framework, where the memory module's weights are updated while the key and value projection matrices remain fixed as hyperparameters. To manage finite memory capacity when processing extremely long sequences, Titans incorporates an adaptive forgetting mechanism using weight decay [1].

This mechanism gradually reduces the influence of older, less surprising information, making room for new inputs. The forgetting gate is data-dependent, meaning the model learns when to clear stale memories based on context rather than simply discarding information based on age. This sophisticated memory management system enables Titans to process sequences of millions of tokens while maintaining computational efficiency [1][9].

Persistent Memory Module

The persistent memory module consists of learnable, data-independent parameters that encode task-specific knowledge. These parameters function similarly to the initial tokens in a sequence, providing context-independent information that helps guide the model's processing. Persistent memory addresses a known issue with causal attention: the implicit bias toward initial tokens in the sequence [1].

By introducing learnable parameters at the start of the sequence, the model can redistribute attention weights more effectively, improving overall performance. From a memory perspective, persistent memory stores abstractions of task knowledge, enabling better task mastery. From a feedforward network perspective, these parameters function like attention weights in Transformer feedforward layers, acting similarly to key-value pairs but independent of input [1][10].

ARCHITECTURAL VARIANTS OF TITANS

Memory as Context (MAC)

The Memory as Context (MAC) variant treats the long-term memory as additional context for the attention mechanism. In this design, the model retrieves relevant information from the neural memory module and concatenates it with the current input sequence before applying attention. This allows the attention mechanism to directly attend to both current and historical information, making informed decisions about what to store in memory and what to retrieve [1].

The MAC architecture segments long sequences into fixed-size chunks. For each incoming segment, the model queries the long-term memory to retrieve relevant historical context, combines this with persistent memory parameters, and processes the concatenated sequence through attention. The attention output is then used to update the long-term memory for the next segment. This approach has three key advantages: (1) attention can decide whether historical information is relevant to the current context; (2) attention helps filter which information from the current segment should be stored in memory; and (3) different components operate at test time with distinct roles [1].

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

Memory as Gate (MAG)

The Memory as Gate (MAG) variant combines the outputs of short-term and long-term memory through a gating mechanism. In this design, sliding window attention processes the immediate context while the neural memory module provides historical information. A gating function dynamically blends these two streams, controlling the contribution of each based on the current input [1].

The gating mechanism operates in parallel, with both memory systems processing the input simultaneously. This design is particularly effective for tasks requiring dynamic integration of short-term precision and long-term context, such as time-series forecasting and speech processing. The gating parameters are learned during training, allowing the model to adaptively balance the contributions of different memory systems for specific tasks [1][3].

Memory as Layer (MAL)

The Memory as Layer (MAL) variant implements the neural memory module as a distinct layer in the network architecture. In this design, the input first passes through the memory module, which compresses and encodes both past and current information, before being processed by the attention mechanism. This sequential arrangement is conceptually simpler but limits the ability of the two memory systems to interact dynamically [1].

A special case of MAL is the Long-term Memory Module (LMM) without attention, where the neural memory operates as a standalone sequence model. This variant demonstrates that the long-term memory module can function independently as a powerful recurrent architecture, validating the design principle that different memory systems should be capable of operating autonomously [1][4].

TRAINING AND OPTIMIZATION

Titans employs a meta-learning framework where the neural memory module learns in an inner loop while the overall architecture is trained in an outer loop. During training, the memory module's weights are updated via gradient descent on the associative memory loss, while the projection matrices and other architectural parameters are optimized through standard backpropagation. This nested optimization enables the memory module to learn how to memorize effectively while the outer loop learns how to utilize the memory system [1][13].

A key innovation in Titans is the tensorization of the gradient descent update, which enables efficient parallel training on GPUs and TPUs. By restructuring the memory update operations to use matrix multiplications, Titans achieves training throughput comparable to modern recurrent models while maintaining the expressiveness of deep neural memory. The training process incorporates momentum in the surprise measure and weight decay for forgetting, both of which contribute significantly to performance [1].

The parallelization strategy involves dividing the sequence into chunks and applying mini-batch gradient descent within each chunk. When parameters are time-invariant within chunks, the system behaves as a linear time-invariant system, which can be computed efficiently using global convolutions. This approach enables Titans to scale to sequences of over 2 million tokens while maintaining competitive training speed [1][11].

COMPARATIVE ANALYSIS WITH EXISTING ARCHITECTURES

Table 1 presents a comprehensive comparison of Titans with other prominent neural architectures for sequence modeling. The comparison highlights key differences in computational complexity, memory requirements, and capabilities across different context lengths.

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

Architecture	Complexity	Memory	Context Length	Test-Time Learning
Transformer	$O(n^2)$	O(n)	Limited	No
LSTM	O(n)	O(1)	Long	No
Mamba	O(n)	O(1)	Very Long	No
RWKV	O(n)	O(1)	Very Long	No
Titans	O(n)	O(1)	2M+ tokens	Yes

Table 1: Comparison of Neural Architectures for Sequence Modeling

As shown in Table 1, Titans achieves a unique combination of properties that distinguishes it from existing architectures. Unlike standard Transformers with their quadratic complexity, Titans maintains near-linear scaling with sequence length. Compared to recurrent models like LSTM and state space models like Mamba, Titans offers superior expressiveness through its deep neural memory module while maintaining comparable computational efficiency [1][3][4][11].

The RWKV architecture shares Titans' goal of combining Transformer-like training efficiency with RNN-like inference efficiency. However, Titans' test-time learning capability and surprise-based memory management provide a more dynamic and adaptive approach to long-term memory. While RWKV achieves impressive results through its linear attention mechanism, Titans' neural memory module offers greater flexibility in what information is stored and how it is retrieved [1][11].

FUTURE RESEARCH DIRECTIONS

The introduction of Titans opens several promising avenues for future research. First, the integration of Titans' memory mechanisms with other architectural innovations, such as mixture-of-experts (MoE) models and multi-modal architectures, presents exciting opportunities for scaling both model capacity and context length simultaneously [1][10].

Second, the theoretical understanding of test-time learning in neural networks remains incomplete. Further research is needed to characterize the convergence properties, generalization bounds, and failure modes of meta-learning-based memory systems. The connection between surprise-based memory updates and human cognitive processes also warrants deeper investigation [1][14][15].

Third, the application of Titans to specific domains such as code generation, scientific literature analysis, and multi-document reasoning presents both technical challenges and significant potential impact. Adapting the surprise metric and memory management strategies for domain-specific characteristics could yield substantial improvements in practical applications [1][12].

Finally, the development of more efficient hardware implementations and specialized kernels for Titans' memory operations could further improve its computational efficiency. The tensorization strategies employed in Titans suggest opportunities for co-design with hardware accelerators to maximize throughput for long-context applications [1][11].

myresearchgo Volume 2, Issue 4, April 2026, ISSN: 3107-3816 (Online)

CONCLUSION

Titans represents a paradigm shift in neural architecture design, introducing a neural long-term memory module that learns to memorize at test time. By combining attention-based short-term memory with deep neural long-term memory and persistent task-specific memory, Titans creates a hybrid architecture that addresses the limitations of both Transformers and recurrent models. The three architectural variants MAC, MAG, and MAL provide flexible options for incorporating memory into different application scenarios, each with distinct trade-offs between computational efficiency and modeling power [1].

The theoretical foundations of Titans, grounded in human cognitive memory systems and meta-learning principles, provide a principled framework for developing more capable and efficient sequence models. The surprise metric, adaptive forgetting mechanisms, and tensorized training procedures demonstrate how insights from psychology and optimization theory can be translated into practical architectural innovations [1][6][9].

As the field continues to push toward longer context windows and more complex reasoning tasks, the concepts introduced in Titans are likely to influence the next generation of neural architectures. The ability to process millions of tokens with near-constant computational complexity opens new possibilities for applications requiring extensive context understanding, from analyzing entire codebases to conducting comprehensive literature reviews [1][12].

The success of Titans also highlights the importance of continued research into alternative architectural paradigms beyond the Transformer. While attention mechanisms have dominated the field for nearly a decade, the computational constraints of quadratic scaling necessitate innovation. Titans demonstrates that by rethinking fundamental assumptions about how neural networks process and store information, significant advances are still possible [1][3][11].

REFERENCES

[1] A. Behrouz, P. Zhong, and V. Mirrokni, "Titans: Learning to Memorize at Test Time," arXiv preprint arXiv:2501.00663, 2024.

[2] A. Vaswani et al., "Attention Is All You Need," in Proc. NeurIPS, 2017, pp. 5998-6008.

[3] T. Dao and A. Gu, "Mamba-2: State Space Duality," arXiv preprint arXiv:2405.21060, 2024.

[4] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," arXiv preprint arXiv:2312.00752, 2023.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[6] J. Schmidhuber, "Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks," Neural Computation, vol. 4, no. 1, pp. 131-139, 1992.

[7] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing Machines," arXiv preprint arXiv:1410.5401, 2014.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in Proc. NeurIPS, 2014, pp. 3104-3112.

[10] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171-4186.

[11] B. Peng et al., "RWKV: Reinventing RNNs for the Transformer Era," in Proc. EMNLP, 2023, pp. 14558-14582.

[12] X. Wang et al., "Memory-Augmented Large Language Models," arXiv preprint arXiv:2409.00019, 2024.

[13] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," in Proc. ICLR, 2017.

[14] Y. Sun et al., "Test-Time Training with Self-Supervision for Generalization under Distribution Shifts," in Proc. ICML, 2020, pp. 9229-9248.

[15] T. Yeo et al., "Rapid Network Adaptation: Learning to Adapt Neural Networks Using Test-Time Feedback," in Proc. ICCV, 2023, pp. 19681-19691.

[16] Priyanka, & Swami, N. (2026). Stability and data dependence of fixed points in metric spaces relations. myresearchgo, 2(1), 83. https://doi.org/10.64448/myresearchgo..vol.2.issue.1.11

[17] Kumari, A., & Basotia, V. (2026). Numerical methods for solving boundary value problems of the wave equation. myresearchgo, 2(1),74. https://doi.org/10.64448/myresearchgo..vol.2.issue.1.10

[18] L. C. Kasireddy, L. Popuri, G. Karunanithi, A. Varghese, S. Ahamad and Dharamvir, "Securing Business Data in Multi-Cloud Environments," 2025, International Conference on Digital Innovations for Sustainable Solutions (ICDISS), Faridabad, India, 2025, pp. 1-6, https://doi.org/10.1109/ICDISS68238.2025.11320589

[19] L. C. Kasireddy, S. Paruchuri, C. Janakamma, A. Sarawat, K. C. Ravi and R. Kumar Chandu, "Cloud-Oriented IoT: Distributed Power-Aware Security Scheme with Data Integrity and Performance Enhancement," 2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 2025, pp. 1-6, https://doi.org/10.1109/WorldSUAS66815.2025.11199185

[20] L. C. Kasireddy, A. Jeraldine Viji, P. K. Sholapurapu, D. Sowjanya Kolluru, D. u. Vishweshwar and P. Agrawal, "Intelligent Intrusion Detection using Artificial Bee Colony-Based Rule Discovery Techniques,"

2025 IEEE Madhya Pradesh Section Conference (MPCON), Jabalpur,India, 2025, pp. 691-696, https://doi.org/10.1109/MPCON66082.2025.11256592

[21] L. C. Kasireddy, S. Paruchuri, C. Janakamma, A. Sarawat, K. C. Ravi and R. Kumar Chandu, "Cloud-Oriented IoT: Distributed Power-Aware Security Scheme with Data Integrity and Performance Enhancement,"2025 World Skills Conference on Universal Data Analytics and Sciences

(WorldSUAS), Indore, India, 2025, pp. 1-6, https://doi.org/10.1109/WorldSUAS66815.2025.11199185

[22] J. L, L. Chandrakanth Kasireddy, R.V.Palanivel, G.Sushma, K. Bhimaavarapu and P. V. Reddy, "Predictive Modeling in Economics: The Role of AI and Deep Learning," 2025 World Skills Conference on

Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 2025, pp. 1-7, https://doi.org/10.1109/WorldSUAS66815.2025.11199198

[23] N. Soni, L. C. Kasireddy, T. S, C. Sinhgadiya, S. Kumar and A. T S, "A Recurrent Neural Network Framework for Effective DDoS Attack Detection in Cloud Computing," 2025 2nd International Conference On Multidisciplinary Research and Innovations in Engineering (MRIE), Gurugram, India, 2025, pp. 594-598, https://doi.org/10.1109/MRIE66930.2025.11156616