Transformer^2 and Titans : The Mathematics

Miquel Noguer i Alonso
Artificial Intelligence Finance Institute

January 17, 2025

Abstract

Large Language Models (LLMs) face critical challenges in balancing task-specific adaptation, computational efficiency, and contextual flexibility. This paper offers an in-depth mathematical and comparative analysis of two pioneering frameworks, Transformer Squared Sun et al. [2025] and Titans Behrouz et al. [2024], which address these challenges through novel methodologies. Transformer Squared introduces a self-adaptive architecture based on Singular Value Decomposition (SVD), enabling dynamic modification of weight matrices via learnable expert vectors, optimized with reinforcement learning. This rank-preserving, parameter-efficient approach allows real-time adaptation to unseen tasks while maintaining minimal computational overhead. Titans, on the other hand, advances the concept of neural memory through biologically inspired mechanisms, integrating long-term and short-term memory systems with adaptive forgetting and contextual encoding.

1 Transformer^2: Advanced Mathematical Formulation

1.1 Foundational Mathematical Framework

Definition 1 (Adaptive Weight Space Transformation)**.** *Let $\mathcal{W}$ be the neural network weight space defined as:*

$$\mathcal{W} = \left\{ W \in \mathbb{R}^{m \times n} : \textit{rank}(W) = k,\ m, n \in \mathbb{N},\ k \leq \min(m, n) \right\}$$

We define the adaptive transformation $\Phi : \mathcal{W} \times \mathbb{R}^r \to \mathcal{W}$ *as the mapping:*

$$\Phi(W, z) = U\mathbf{\Sigma}'V^\top \tag{1}$$

where:

- $W = U\mathbf{\Sigma}V^\top$ *is the singular value decomposition*
- $\mathbf{\Sigma}' = \mathbf{\Sigma} \odot \mathit{diag}(z)$
- $z \in \mathbb{R}^r$ *is a learnable adaptation vector*
- $\odot$ *denotes element-wise Hadamard product*

Theorem 1 (Adaptive Rank Preservation)**.** *For any* $W \in \mathcal{W}$ *and* $z \in \mathbb{R}^r$*, the mapping* $\Phi(W, z)$ *satisfies:*

1. $\mathit{rank}\big(\Phi(W, z)\big) = \mathit{rank}(W)$
2. $\|\Phi(W, z) - W\|_F \leq \epsilon$
3. $\mathit{dist}\big(\mathit{span}(W), \mathit{span}\big(\Phi(W, z)\big)\big) \leq \delta$

where $\epsilon, \delta > 0$ *depend on the magnitude of* z*.*

Proof. Consider the singular value decomposition $W = U\mathbf{\Sigma}V^\top$.

1) **Rank Preservation:**

$$\begin{aligned}\operatorname{rank}\big(\Phi(W, z)\big) &= \operatorname{rank}\big(U\operatorname{diag}(z \odot \sigma(W))V^\top\big) \\ &= \operatorname{rank}\big(\operatorname{diag}(z \odot \sigma(W))\big) \\ &= \operatorname{rank}(W)\end{aligned}$$

2) **Frobenius Norm Bound:**

$$\begin{aligned}\|\Phi(W, z) - W\|_{\mathrm{F}} &= \big\|U\operatorname{diag}(z \odot \sigma(W))V^\top - U\mathbf{\Sigma}V^\top\big\|_{\mathrm{F}} \\ &\leq \|U\|_2\|V\|_2\|\operatorname{diag}(z \odot \sigma(W)) - \mathbf{\Sigma}\|_{\mathrm{F}}\end{aligned}$$

3) **Subspace Distance:** By the Davis-Kahan theorem, the distance between principal subspaces is bounded by the perturbation of singular values. □

1.2 Probabilistic Adaptation Framework

Definition 2 (Stochastic Weight Adaptation)**.** *The probabilistic weight adaptation is defined as a Gaussian distribution:*

$$p(W' \mid W, z) = \mathcal{N}\big(\Phi(W, z), \sigma^2 I\big) \tag{2}$$

where σ^2 controls the adaptation variance.

1.3 Reinforcement Learning Formulation

Definition 3 (Adaptive Policy Optimization)**.** *The adaptation objective is a constrained optimization problem:*

$$\begin{aligned} \max_{\theta_z} \quad & \mathbb{E}_{\pi_{\theta_{W'}}}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] \\ \textit{subject to} \quad & D_{KL}\big(\pi_{\theta_{W'}} \| \pi_{\theta_W}\big) \leq \epsilon \\ & \|\theta_z\|_2 \leq \delta \end{aligned} \tag{3}$$

where:

- $\pi_{\theta_{W'}}$ *is the adapted policy*
- $r(s_t, a_t)$ *is the reward function*
- D_{KL} *is the Kullback-Leibler divergence*
- $\epsilon, \delta > 0$ *are constraint bounds*

1.4 Computational Complexity Analysis

Theorem 2 (Computational Complexity Characterization)**.** *The adaptive transformation $\Phi(W, z)$ exhibits the following complexity:*

1) ***Time Complexity:***

$$T(\Phi) = \mathcal{O}(rmn + r^2) \tag{4}$$

2) ***Space Complexity:***

$$S(\Phi) = \mathcal{O}(r) \tag{5}$$

3) ***Information Theoretic Complexity:***

$$I(\Phi) = \mathcal{O}(r \log r) \tag{6}$$

1.5 Expert Vector Interpolation

Definition 4 (Expert Vector Mixture)**.** *Define the expert vector mixture space as:*

$$\mathcal{Z} = \left\{ z' = \sum_{k=1}^{K} \alpha_k z_k : \sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0 \right\} \tag{7}$$

Theorem 3 (Expert Vector Interpolation Properties)**.** *The expert vector mixture satisfies:*

1. ***Convexity:*** $\Phi(W, z') \in conv\big(\{\Phi(W, z_k)\}_{k=1}^{K}\big)$
2. ***Bounded Variation:***

$$\|\Phi(W, z') - \Phi(W, z)\|_F \leq C\|\alpha - \beta\|_2 \tag{8}$$

1.6 Convergence Analysis

Theorem 4 (Adaptation Convergence)**.** *The adaptation process converges with the following guarantee:*

$$\mathbb{E}\left[\|\nabla J(\theta_z)\|^2\right] \leq \frac{C}{\sqrt{T}} \tag{9}$$

where:

- $J(\theta_z)$ *is the adaptation objective*
- T *is the number of iterations*
- $C > 0$ *is a problem-dependent constant*

This comprehensive mathematical treatment provides a rigorous foundation for understanding the Transformer^2 adaptive framework, emphasizing its theoretical underpinnings and computational properties.

2 Titans: Advanced Mathematical Formulation of Neural Long-Term Memory

2.1 Foundational Memory Theoretical Framework

Definition 5 (Neural Memory State Space)**.** *Consider a memory state space* $\mathcal{M} = \{M \in \mathbb{R}^{d \times d} : M \text{ represents memory configuration}\}$*, where* d *is the memory dimension.*

Define the memory state evolution operator $\Psi : \mathcal{M} \times \mathbb{R}^n \to \mathcal{M}$ *as:*

$$\Psi(M_{t-1}, x_t) = (1 - \alpha_t)M_{t-1} + S_t$$

where:

- M_{t-1} *is the previous memory state*
- x_t *is the current input*
- $\alpha_t \in [0, 1]$ *is the memory forgetting rate*
- S_t *is the surprise-based memory update term*

2.2 Surprise Metric Formalization

Definition 6 (Contextual Surprise Metric)**.** *The surprise term* S_t *is defined as a compositional momentum-based update:*

$$S_t = \eta_t S_{t-1} - \theta_t \nabla \ell(M_{t-1}; x_t)$$

where:

- η_t *is a data-dependent surprise decay factor*
- θ_t *controls momentary surprise incorporation*
- $\ell(M_{t-1}; x_t)$ *is the associative memory loss function*

Theorem 5 (Surprise Metric Properties)**.** *The surprise metric* S_t *satisfies the following properties:*

1. ***Bounded Variation:*** $\|S_t\| \leq C \cdot \max(1, \|S_{t-1}\|)$
2. ***Momentum Preservation:*** $\mathbb{E}[S_t] = (1 - \theta_t)\mathbb{E}[S_{t-1}]$
3. ***Information Retention:*** $Var(S_t) \leq (1 + \eta_t)\, Var(S_{t-1})$

2.3 Associative Memory Formulation

Definition 7 (Associative Memory Loss)**.** *The memory loss function is defined as:*

$$\ell(M_{t-1}; x_t) = \|M_{t-1}(k_t) - v_t\|_2^2$$

where:

- $k_t = x_t W_K$*: Key projection*
- $v_t = x_t W_V$*: Value projection*
- W_K, W_V*: Learnable projection matrices*

2.4 Memory Retrieval Mechanism

Definition 8 (Memory Retrieval Operator)**.** *Define the memory retrieval function* $\Gamma : \mathcal{M} \times \mathbb{R}^n \to \mathbb{R}^n$*:*

$$y_t = \Gamma(M^*, q_t), \quad q_t = x_t W_Q$$

where:

- M^* *is the optimized memory state*
- q_t *is the query projection*
- W_Q *is the query projection matrix*

2.5 Memory Incorporation Strategies

2.5.1 Memory as Context (MAC)

Definition 9 (Contextual Memory Integration)**.** *The MAC strategy defines memory integration as:*

$$\begin{aligned} h_t &= M_{t-1}^*(q_t) \\ \tilde{S}^{(t)} &= [p_1, \ldots, p_{N_p} \| h_t \| S^{(t)}] \\ y_t &= Attn(\tilde{S}^{(t)}) \end{aligned}$$

2.5.2 Memory as Gate (MAG)

Definition 10 (Gated Memory Integration). *The MAG strategy implements:*

$$y = SW\text{-}Attn^*(\tilde{x})$$
$$o = y \otimes M(\tilde{x})$$

2.5.3 Memory as Layer (MAL)

Definition 11 (Layer-based Memory Integration). *The MAL strategy defines:*

$$y = M(\tilde{x})$$
$$o = SW\text{-}Attn(y)$$

2.6 Theoretical Convergence Analysis

Theorem 6 (Memory State Convergence). *The memory state M_t converges with the following properties:*

$$\|\nabla \ell(M_t)\| \leq O\left(\frac{1}{\sqrt{t}}\right)$$

under the following conditions:

- *Bounded gradients*
- *Appropriate learning rates*
- *Continuous memory state updates*

2.7 Computational Complexity

Theorem 7 (Computational Complexity Characterization). *The Titans memory module exhibits:*

1) Memory Space Complexity:

$$Space(Memory) = O(d^2)$$

2) Update Time Complexity:

$$Time(Update) = O(N \times d^2)$$

Where d is the memory dimension and N is the number of update steps.

2.8 Probabilistic Memory Representation

Definition 12 (Stochastic Memory State)**.** *Define the memory state as a probabilistic distribution:*

$$p(M_t|M_{t-1}, x_t) = \mathcal{N}(\mu_t, \Sigma_t)$$

where:

- μ_t *is the mean memory update*
- Σ_t *represents the uncertainty in memory state*

This comprehensive mathematical treatment provides a rigorous theoretical foundation for the Titans neural long-term memory approach, highlighting its innovative memory management and adaptation mechanisms.

3 Comprehensive Comparative Analysis

3.1 Theoretical Foundations

3.1.1 Transformer^2

The core innovation lies in Singular Value Decomposition (SVD) of weight matrices, allowing selective modification through:

- Decomposing weight matrices into singular vectors and values
- Learning an adaptation vector to scale singular values
- Preserving matrix rank while enabling targeted modifications

3.1.2 Titans

The approach centers on a neural memory module characterized by:

- Surprise-based memory updates
- Adaptive memorization of contextual information
- Dynamic memory management through forgetting mechanisms

Aspect	Transformer^2	Titans
Core Innovation	Singular Value Fine-Tuning	Neural Long-Term Memory
Adaptation Strategy	Two-pass expert vector mixing	Contextual memory encoding
Optimization Approach	Reinforcement Learning with KL divergence	Surprise-based memory update
Computational Complexity	O(rmn + r^2)	O(sequence length)
Flexibility Mechanism	Singular value scaling	Dynamic memory encoding
Memory Handling	Expert vector interpolation	Adaptive memory state management

3.2 Adaptation Mechanisms

3.3 Detailed Comparative Insights

3.3.1 Adaptation Strategies

- **Transformer^2:**
  1. Prompt Engineering
  2. Classification Expert
  3. Few-Shot Adaptation via Cross-Entropy Method
- **Titans:**
  1. Memory as Context (MAC)
  2. Memory as Gate (MAG)
  3. Memory as Layer (MAL)

3.3.2 Theoretical Motivations

- **Transformer^2** draws inspiration from:
  - Computational efficiency
  - Parameter-efficient fine-tuning
  - Dynamic task adaptation
- **Titans** is motivated by:
  - Cognitive memory systems

- Biological learning mechanisms
- Context-aware information processing

3.4 Performance Characteristics

3.4.1 Strengths of Transformer²

- Mathematically rigorous adaptation approach
- Minimal parameter overhead
- Efficient across diverse task domains
- Preserves model rank during adaptation

3.4.2 Strengths of Titans

- Flexible memory management
- Adaptive context understanding
- Biological-inspired learning mechanism
- Robust to long-context challenges

4 Conclusion

Transformer^2 and Titans represent pivotal advances in addressing the fundamental challenges of large language model adaptation. Transformer^2 introduces a mathematically sophisticated framework leveraging Singular Value Decomposition, offering rank-preserving weight matrix transformations and reinforcement learning-driven expert vector optimization. This approach provides a principled mechanism for selective model modification with minimal computational overhead.

Titans presents a more radical reimagining of model adaptation, drawing inspiration from cognitive neuroscience. Its neural long-term memory architecture fundamentally transforms how models process and retain information. By implementing a sophisticated surprise-based memory update mechanism, Titans creates a dynamic memory system that adaptively encodes contextual information. The approach goes beyond traditional static

memory models, introducing three innovative memory incorporation strategies: Memory as Context (MAC), Memory as Gate (MAG), and Memory as Layer (MAL). These strategies allow the model to dynamically manage its memory state, selectively forget irrelevant information, and adaptively respond to contextual nuances in a manner reminiscent of biological learning processes.

Both methodologies challenge traditional static fine-tuning paradigms, demonstrating innovative approaches to dynamic model modification. They represent a critical evolution in machine learning architectures, enabling more intelligent systems capable of contextual adaptation and nuanced internal representation modification. By providing principled strategies for task-specific flexibility, these approaches mark a significant step towards more responsive and intelligent computational models.

References

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time, 2024. URL https://arxiv.org/abs/2501.00663.

Qi Sun, Edoardo Cetin, and Yujin Tang. Transformer^2: Self-adaptive llms, 2025. URL https://arxiv.org/abs/2501.06252.