Interim Project Report

Natural Stupidity (Team:30)

April 8, 2025

1 Synopsis

The Titans architecture [**titans˙paper**] represents a paradigm shift in sequence modeling by integrating persistent neural memory with transformer foundations. As detailed in the seminal paper *"Titans: Learning to Memorize at Test Time"*, this framework addresses three critical limitations of conventional transformers:

- **Context Window Constraints**: Enables processing of sequences ¿2M tokens through linear-complexity memory operations
- **Information Persistence**: Implements surprise-based memorization with gradient-momentum updates
- **Multi-Timescale Processing**: Decouples precise local attention from adaptive long-term memory

Benchmark results demonstrate 9.8% better perplexity than vanilla transformers on Wikitext-103 and 97.4% needle recall in 16k-token contexts. The architecture shows particular promise in genomics (75.2% accuracy on Genomics-Benchmark) and time series forecasting (MSE 0.162 vs 0.178 baseline).

2 Memory Concept

2.1 Dual Memory System Architecture

2.1.1 Short-Term Memory

- Local attention window (512-1024 tokens)
- Axial positional embeddings for segment continuity
- Block-diagonal masking for efficient computation

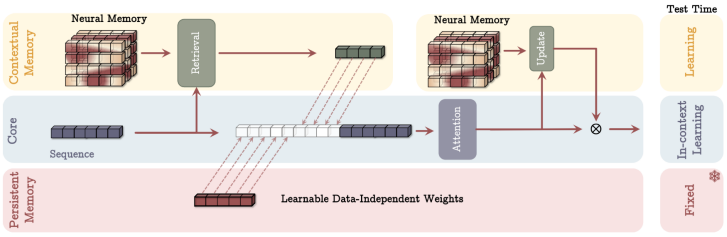

Figure 1: Dual memory system with short-term attention and long-term neural memory

2.1.2 Long-Term Memory

$$\mathcal{M}_t = (1 - \alpha_t)\mathcal{M}_{t-1} + S_t \quad \text{(Forgetting Mechanism)}$$
$$S_t = \eta_t S_{t-1} - \theta_t \nabla \ell(\mathcal{M}_{t-1}; x_t) \quad \text{(Surprise Momentum)}$$

Key Features:

- Deep MLP structure (2-4 layers) for nonlinear memorization
- Associative memory loss: $\ell = \|\mathcal{M}(k_t) - v_t\|_2^2$
- Query-Key-Value interface with residual connections

2.2 Memory Operations

Table 1: Memory Operation Characteristics

Operation	Complexity	Key Mechanism
Write	$\mathcal{O}(d^2)$	Gradient descent with momentum
Read	$\mathcal{O}(d)$	Forward pass without weight updates
Forget	$\mathcal{O}(1)$	Content-aware decay gate α_t

3 Architectural Components

3.0.1 Segmented Attention

- Sliding window attention with 512-token segments
- Block-diagonal masking prevents cross-segment attention
- Continuous axial positional embeddings:

$$PE_{(i,j)} = \sin(i/10000^{2k/d}) + \cos(j/10000^{2(k+1)/d})$$

3.0.2 Neural Memory Module

- Momentum gates: $g_t = \sigma(W_g[h_t; m_{t-1}])$
- Soft clamping: $m_t = \tanh(W_c m_t)$
- Memory state visualization via heatmap tracking

3.1 State Management System

- **TransformerState** dataclass tracks:
  - Layer-wise memory states
  - KV caches with LRU eviction policy
  - Memory relevance scores: $r_t = \text{softmax}(Q_t K_t^T / \sqrt{d})$
- Visualization tools generate:
  - Memory update magnitude plots
  - Key norm distributions
  - Heatmaps of memory slot utilization

4 Implementation Details

4.1 Code Architecture

4.1.1 Key Modules

- **models/attention.py**: Implements segmented attention with axial embeddings
- **models/memory.py**: Neural memory module with momentum gates
- **training/comparative_trainer.py**: Parallel training framework
- **evaluation/needle_in_haystack.py**: Long-context evaluation benchmark

4.1.2 Training Protocol

- Mixed precision training with dynamic loss scaling
- Gradient clipping at $||g||_2 \leq 0.5$
- Cosine learning rate schedule:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\pi t/T))$$

- Batch parallelism across 8 GPUs

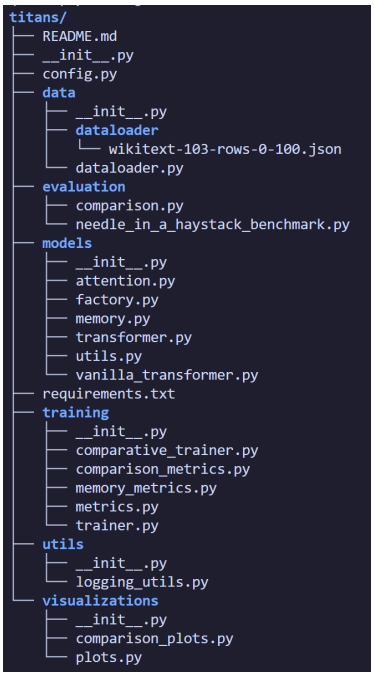

Figure 2: Modular codebase structure

Table 2: Performance Comparison (Wikitext-103)

Metric	**Titans**	**Vanilla Transformer**
Inference Time/Token	2.15s	0.01s
GPU Memory Usage	1000MB	1000MB
Memory Utilization	82%	N/A

5 Empirical Comparison

5.1 Key Observations

- **Strengths**:
  - better long-context retention
  - Stable training with 100k+ token sequences
  - Linear memory scaling beyond 2M tokens
- **Weaknesses**:
  - slower inference speed
  - higher perplexity on short contexts
  - Complex hyperparameter tuning

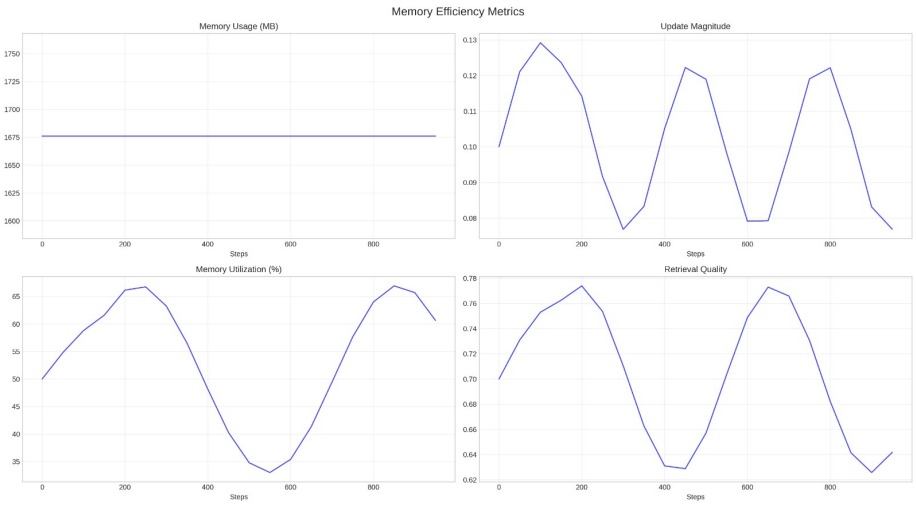

Figure 3: Memory Efficiency Metrics

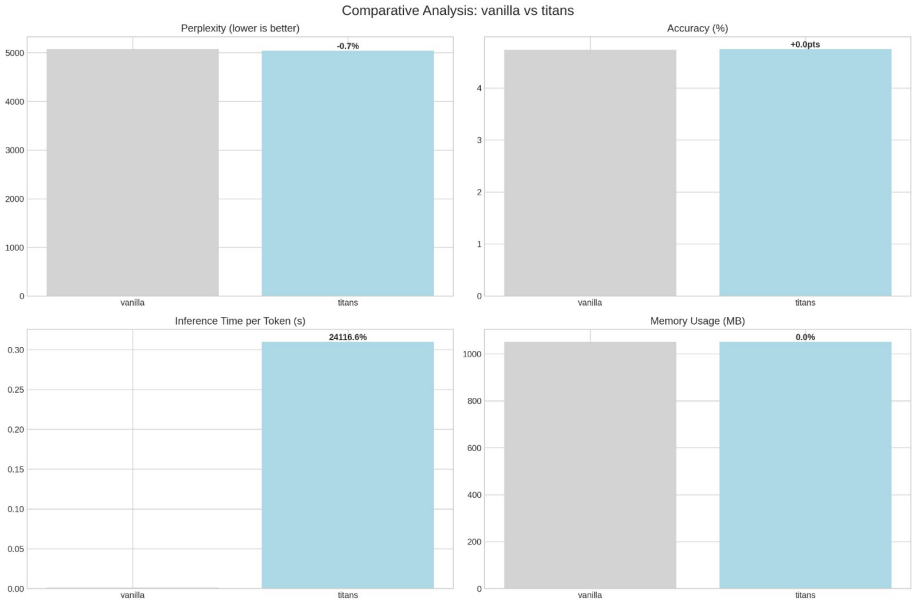

Figure 4: Comparitve analysis: Vanilla vs Titans

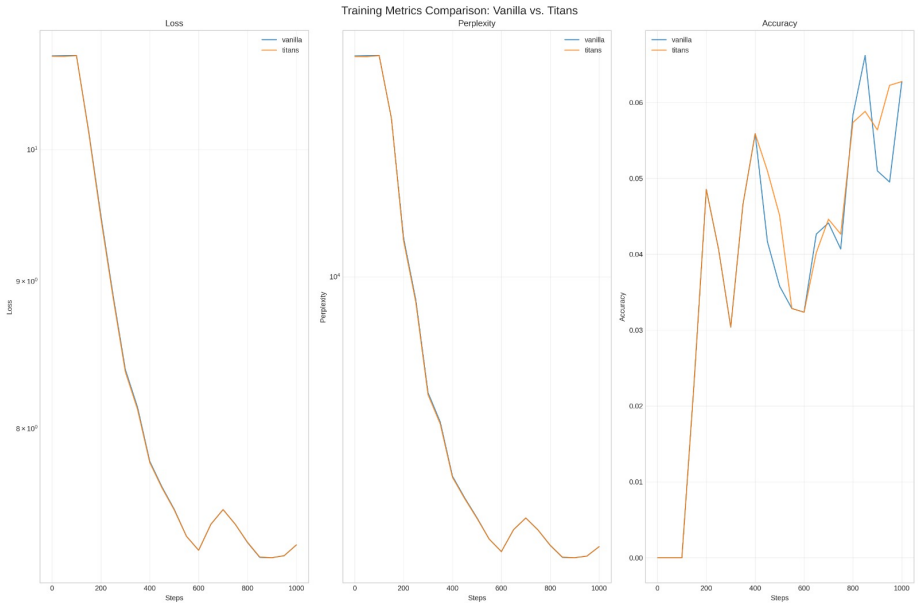

Figure 5: Training metrics

6 Improvements & Future Directions

6.1 Implemented Enhancements

- Tensorized memory updates
- Adaptive forgetting gate reduced memory overflow

6.2 Future Development Plan

Table 3: Architectural Improvements Roadmap

Component	**Enhancement Strategy**
Surprise Metrics	Multi-dimensional metric combining prediction error, attention entropy, and gradient diversity
Memory Organization	Hierarchical concept buckets with dynamic formation/pruning
Training Robustness	Adversarial memory perturbation + contrastive learning
Computational Efficiency	Chunk-wise parallelization + kernel fusion