Titans: Learning to Memorize at Test Time

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni

Google Research

My Two Cents, Tim Büchner

04.03.2025

Abstract

- Recurrent models compress data **and** attention mechanisms capture dependencies.
- Attention models face quadratic complexity limits.
- New **neural long-term memory** enhances attention with history.
- **Titans** combine short-term attention and long-term memory (hybrid models)

Key Questions

- **Q1:** What constitutes a good memory structure?
- **Q2:** What is an effective memory update mechanism?
- **Q3:** What is an optimal memory retrieval process?
- **Q4:** How to design an architecture with interconnected memory modules?
- **Q5:** Is a deep memory module needed for long-term storage?

Contributions

- **Neural Memory Module**: A novel deep memory mechanism that learns via **violating expectations** .
- **Memory Update and Retrieval**: Improved strategies for storing, forgetting, and retrieving past information.
- **Titans Architecture**: A family of models integrating short-term and long-term memory (MAC, MAG, MAL)
- **Scalability**: Titans can process over **2M tokens**, outperforming existing models.

Preliminaries

- $x \in \mathbb{R}^{N \times d_{\text{in}}}$ is the input (most often tokens)
- $\mathcal{M}$ is a neural network (neural memory module)(meta-model)
- $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are query, key, and value of the attention mechanism
- $\mathbf{M}$ is the attention mask
- $S^{(i)}$ is the i-th segment of a segmented input x

Preliminaries (1/3): Attention Mechanisms

- Transformers use **self-attention** to model dependencies.
- Computational complexity grows **quadratically** with sequence length.
- **Linear Transformers** use kernel-based approximations for efficiency.
- Useful for efficient inference for linear attention on their **neural memory module**:

$$\mathcal{M}_t = \mathcal{M}_{t-1} + K_t^T V_t$$
$$y_t = Q_t \mathcal{M}_t$$

Preliminaries (2/3): Recurrent Models

- RNNs and LSTMs compress information into **hidden states**.
- In general, the following operations are needed:

$$\mathcal{M}_t = f(\mathcal{M}_{t-1}, x_t) \quad \text{(Write Operation)}$$
$$y_t = g(\mathcal{M}_t, x_t) \quad \text{(Read Operation)}$$

- **State-space models** extend RNNs for long-term dependencies.
- Modern approaches like **Mamba** improve efficiency and scalability.

Preliminaries (3/3): Memory Perspective

- Memory systems in deep learning:
  short-term (attention) vs. **long-term (recurrent)**.
- **Key challenge:** Storing and retrieving long-range dependencies efficiently.
- **Titans** aim to unify these concepts for **better scalability and effectiveness**
- We will now take a look at their ideas!

Memorize at Test Time

Long-term Memory (1/4): Motivation

- Humans remember **surprising** or significant events better
  - More like if our expectations are not met
- Traditional deep learning models struggle with long-term dependencies.
- **Key idea**: A neural memory module that learns to store and retrieve relevant past information.

Long-term Memory (2/4): Surprise-based Learning

- Define a **momentary surprise** as the gradient of loss with respect to input:

$$S_t = -\theta_t \nabla \ell(\mathcal{M}_{t-1}; x_t)$$

- Use **past surprise** to influence memory updates:

$$\mathcal{M}_t = \mathcal{M}_{t-1} + \eta_t S_{t-1} - \theta_t \nabla \ell(\mathcal{M}_{t-1}; x_t)$$

- Acts like gradient descent with momentum, ensuring better memory stability.
- θ_t (information control factor) and η_t (decay factor) should be *data-dependent*
- $\eta_t \to 0$ ignore last event; $\eta_t \to 1$ use last event fully

Long-term Memory (2/4): Surprise-based Learning Objective

- Surprise metric is based on a loss function $\ell(\,.\,;\,.\,)$, which is that the objective that our memory is learning to act as it at test time
- Build a **associative memory** $\rightarrow$ store past data as (key, value) pairs.
- For a given x, transform into key and $value$:

$$k_t = x_t W_K \quad v_t = x_t W_V$$

- The memory should learn the following association between key and value:

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

- Inner-loop and Outer-loop training!

Long-term Memory (3/4): Forgetting Mechanism

- Key challenge: Managing limited memory capacity.
- Adaptive forgetting mechanism:

$$\mathcal{M}_t = (1 - \alpha_t)\mathcal{M}_{t-1} + +\eta_t S_{t-1} - \theta_t \nabla \ell(\mathcal{M}_{t-1}; x_t)$$

- α_t is data-dependent (not really written here in the text...)
- Controls how much past information should be retained.
- Similar to modern recurrent models (e.g., Mamba, DeltaNet).
- Use simple MLP as model architecture $\rightarrow$ more research possible

Long-term Memory (4/4): Memory Retrieval

- Use stored knowledge effectively during inference.
- Retrieve memory by querying stored keys:

$$y_t = \mathcal{M}^*(q_t), \quad q_t = x_t W_Q$$

- Memory retrieves useful historical information dynamically.
- Enables Titans to handle long-term dependencies efficiently.

Persistent Memory (1/3): Concept

- Long-term memory depends on context, but some knowledge should be static.
- **Persistent memory** consists of **learnable, input-independent parameters**.
- Stores **task-specific knowledge** that should not change at test time.

Persistent Memory (2/3): Integration

- Via persistent **learnable** parameters, concatenated with the input sequence:

$$x_{new} = [p_1, p_2, \ldots, p_{N_p}] || x$$

- Allows the model to incorporate static knowledge into processing
    - **input-independent** memory

Persistent Memory (3/3): Benefits

- Acts as a **task-related knowledge base**, improving generalization.
- Mitigates bias in causal attention, preventing over-reliance on early tokens.
- Enhances **in-context learning** by providing a stable foundation for adaptation.

Incorporate Memory

Memory as a Context (1/3)

- Memory is treated as additional context for attention in an existing model.
- Chunk x into segments, treat $S^{(t)}$ as current context, previous as history
- Memory retrieval:

$$h_t = \mathcal{M}^*_{t-1}(S^{(t)} W_Q)$$

- Updated input sequence:

$$\tilde{S}^{(t)} = [p_1, \ldots, p_{N_p}] \| h_t \| S^{(t)}$$

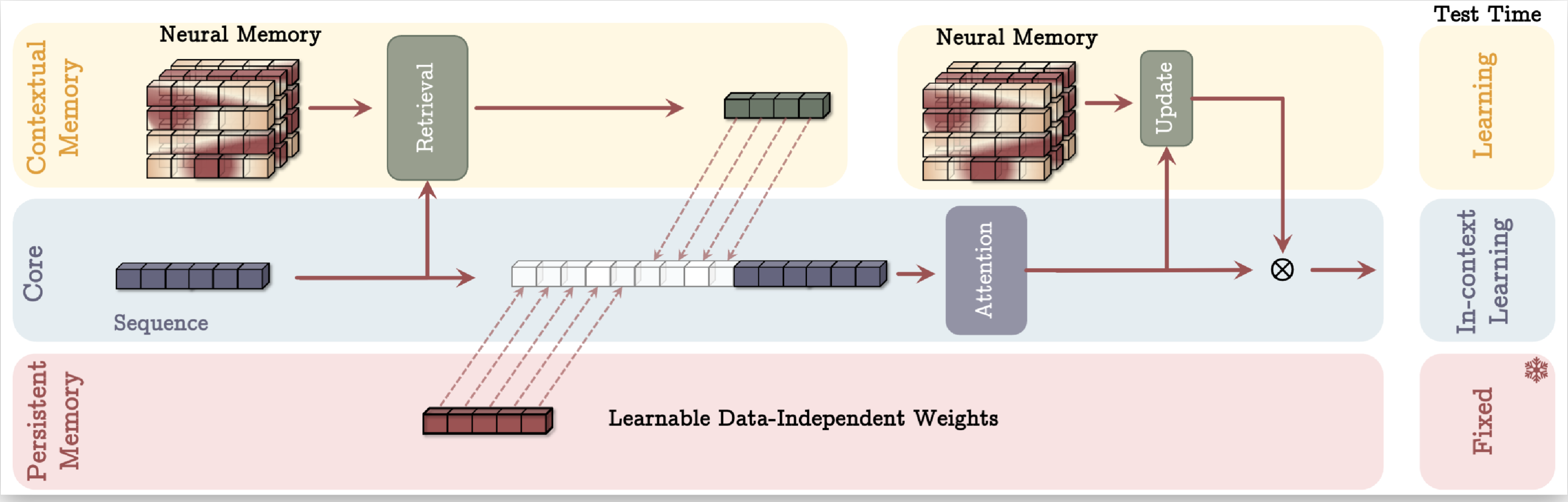
$$y_t = \mathrm{Attn}(\tilde{S}^{(t)})$$

- Update the memory:

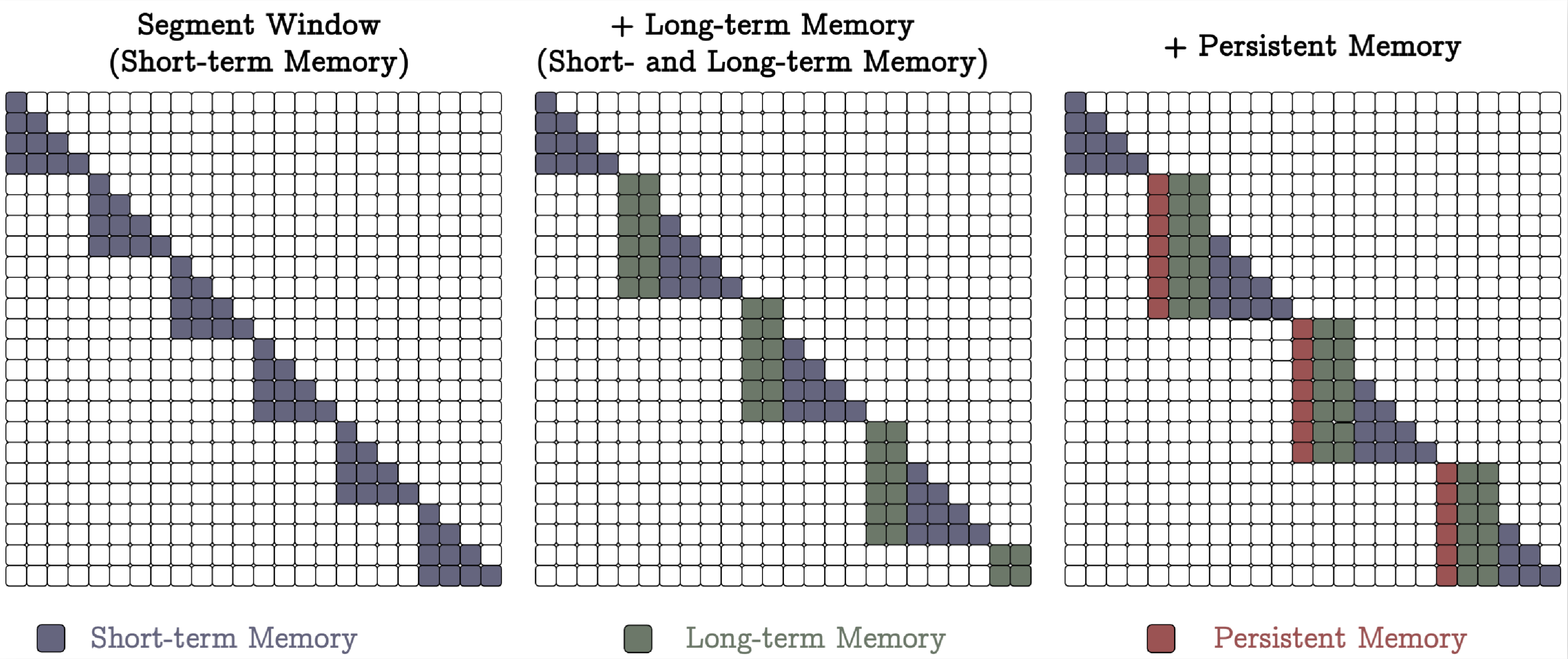
$$\mathcal{M}_t = \mathcal{M}_{t-1}(y_t)$$

$$o_t = y_t \otimes \mathcal{M}^*(y_t)$$

Memory as a Context (2/3)



Memory as a Context (3/3)



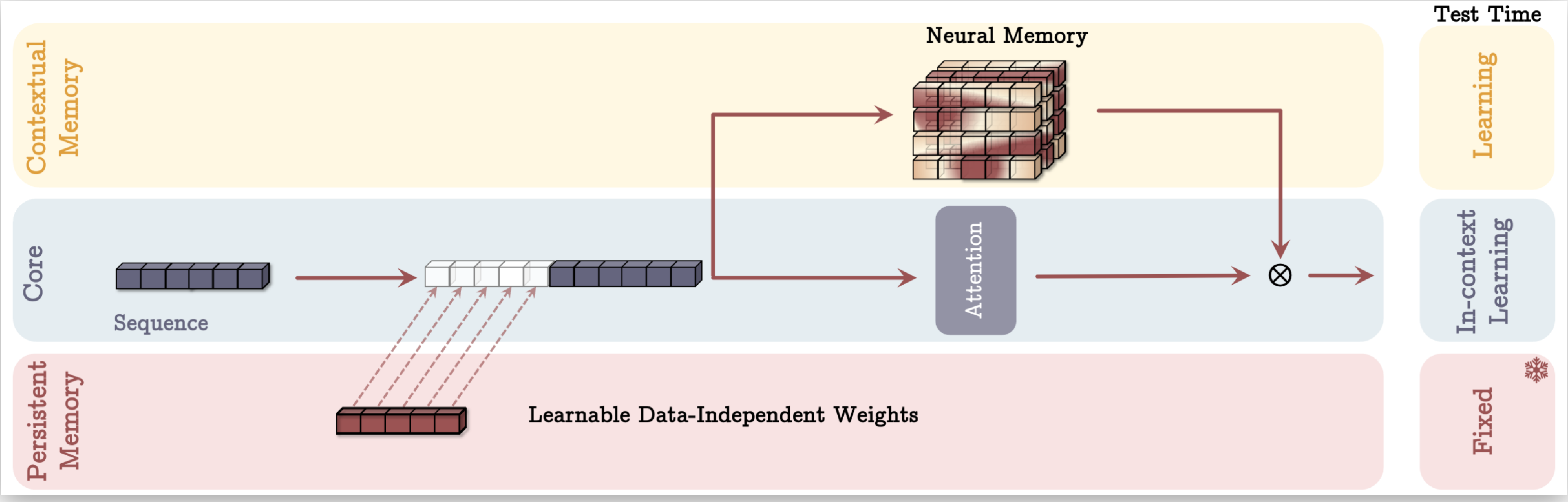
Gated Memory (1/3)

- Use the x to directly update the memory, and use sliding window attention
  - **No segmentation!**
- Do the following updates:

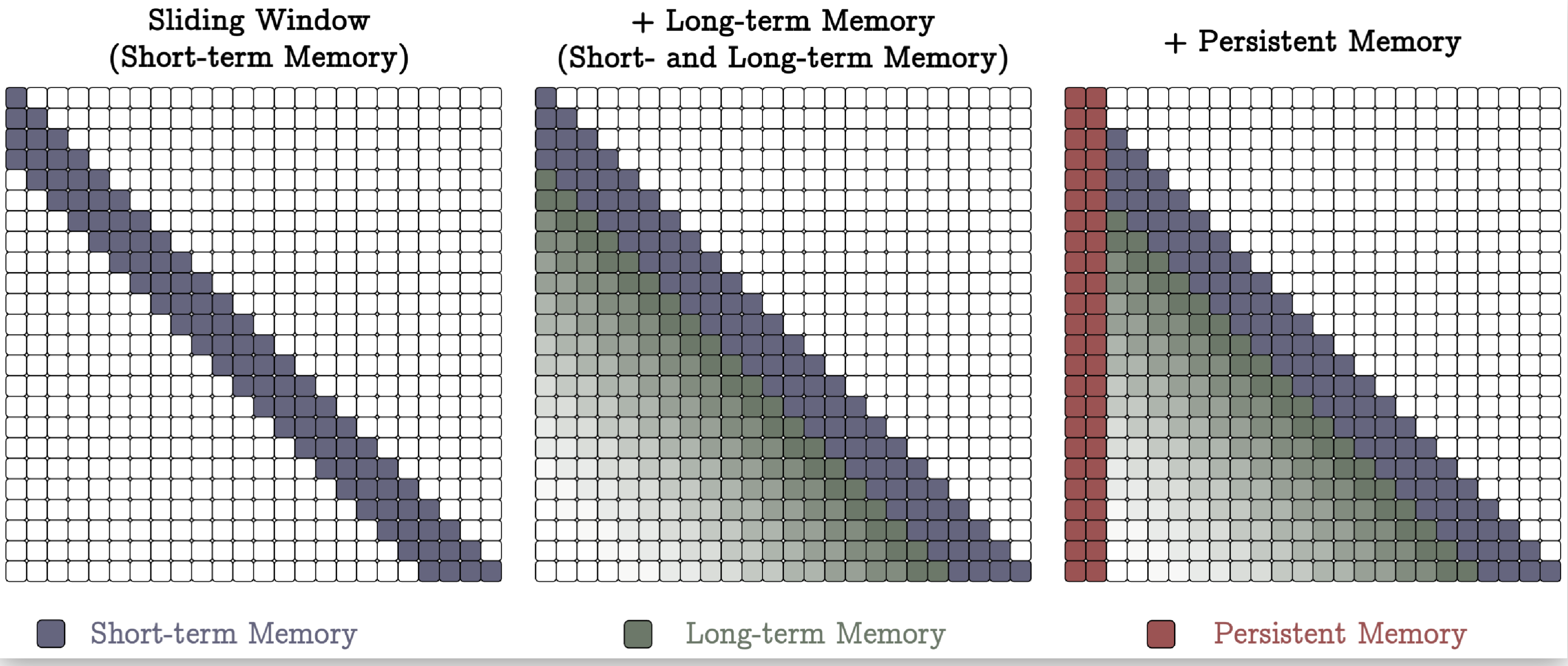
$$\tilde{x} = [p_1, \dots, p_{N_p}] \| x$$
$$y_t = \mathrm{SW\text{-}Attn}(\tilde{x})$$
$$o = y \otimes \mathcal{M}(\tilde{x})$$

- $\otimes$ can be *any* non-linear gating mechanism
  - they normalize y and $\mathcal{M}(\tilde{x})$ (via learnable vectors), and use $\sigma(\cdot)$

Gated Memory (2/3)



Gated Memory (3/3)



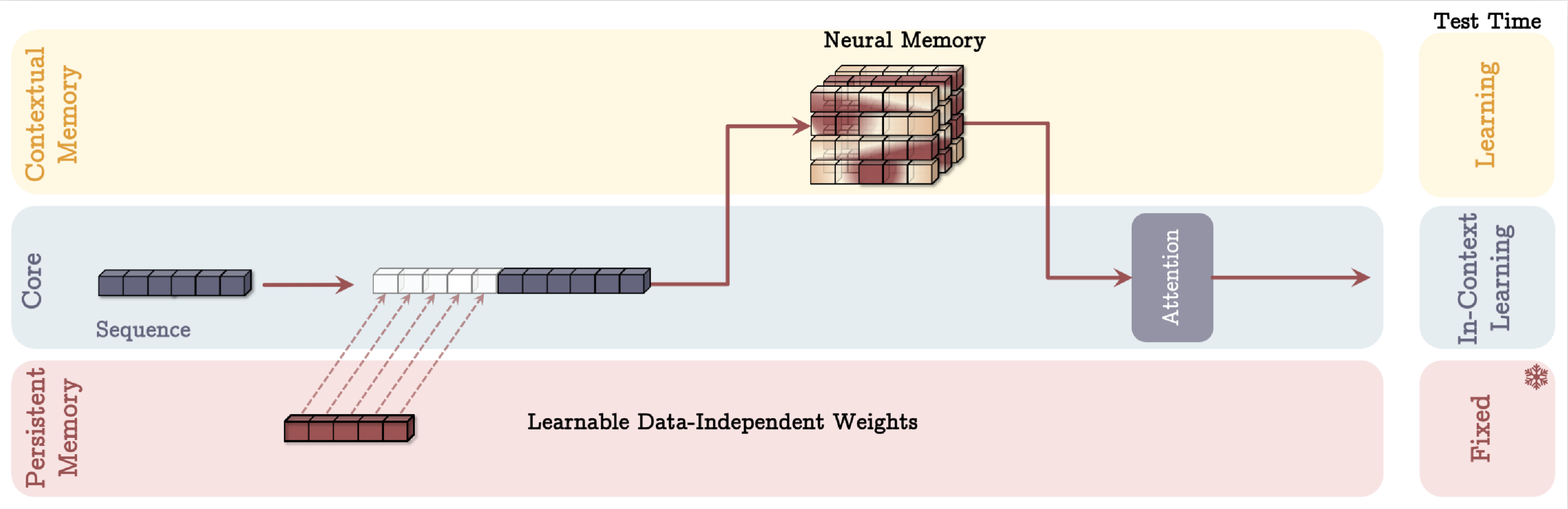
Memory as a Layer (1/2)

- Memory module acts as a separate model layer.
- Do the following updates:

$$\tilde{x} = [p_1, \ldots, p_{N_p}] \| x$$
$$y_t = \mathcal{M}(\tilde{x})$$
$$o = \text{SW-Attn}(y)$$

- No complementary data processing during attention mechanism

Memory as a Layer (2/2)



Implementation Details

- Use residual connections in every block
- Use $\mathrm{SiLU}(\cdot)$ activations function for query, key, and value
- Normalize query and key using ℓ_2-norm
- 1D depthwise-separable convolutions after query, key, value projections

Theorem 4.1.

- Contrary to Transformers, diagonal linear recurrent models, and DeltaNet, all of which are limited to TC^0 (Merrill, Petty, and Sabharwal 2024), Titans are capable of solving problems beyond TC^0, meaning that Titans are theoretically more expressive than Transformers and most modern linear recurrent models in state tracking tasks.

Experiments

- hybrid models are (recurrent + attention) network architectures

Experiments: Language Modeling

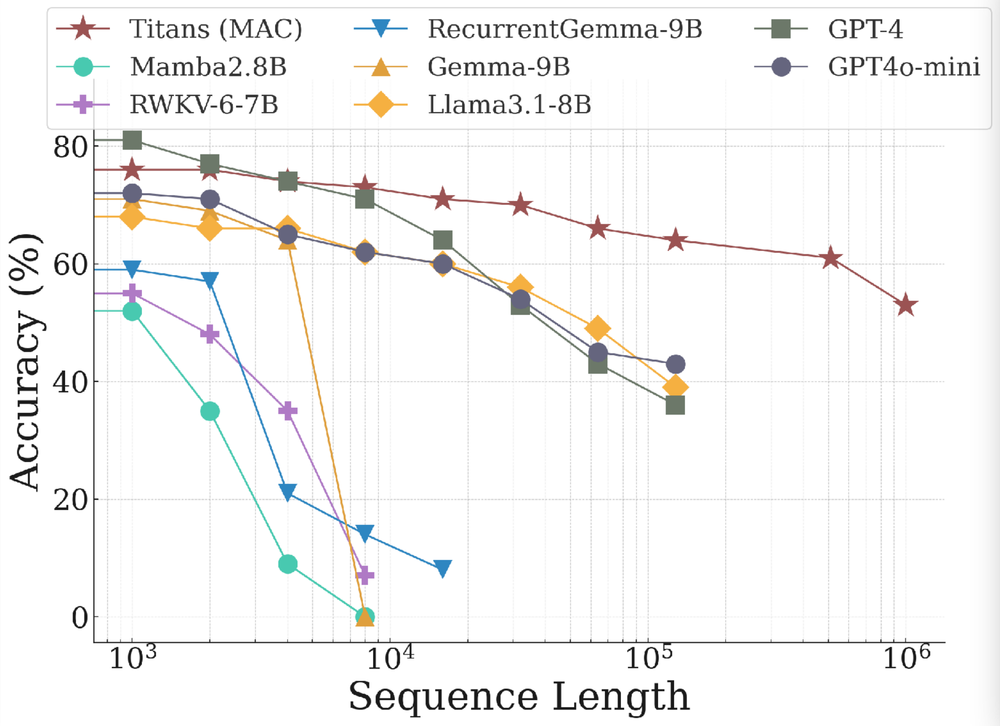
- Evaluated on WikiText and LMB datasets.
- Metric: **Perplexity (lower is better)**.
- **Best model**: Titans generally outperform all others, **hybrid** a bit better
- Outperforms Mamba, Transformer++, and DeltaNet.

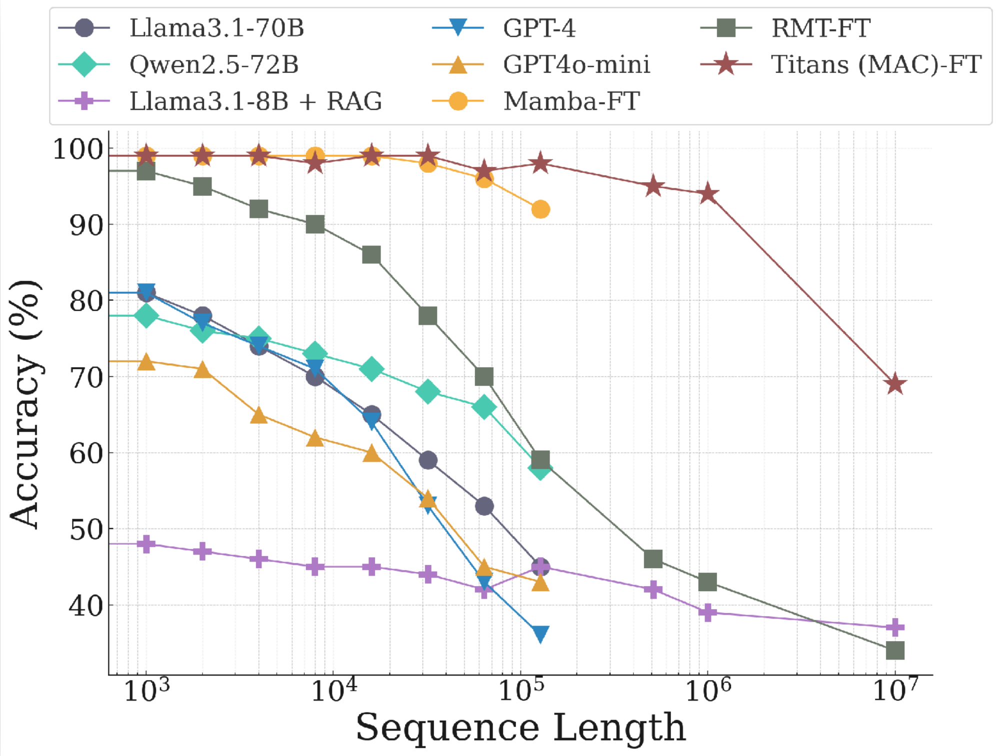
Experiments: Needle in a Haystack

- Evaluated on NIAH benchmark.
- Metric: **Retrieval accuracy** at long sequence lengths (2K, 4K, 8K, 16K)
- **Best model**: Titans (MAC) with **acc = 98.4%** at 16K tokens.
- Demonstrates **effective long-term retrieval** compared to baselines.
    - Outperform clearly at 16K S-NIAH-W

Experiments: BABILong Benchmark

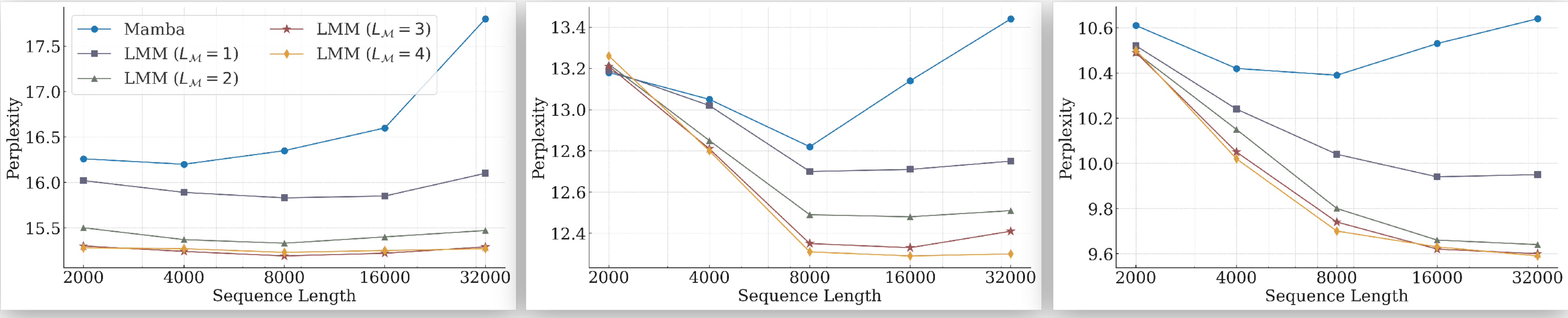
- Not only a single information, but several important factors *hidden*
  - used for *extremly* long documents
- Few-shot and fine-tuning settings compared





Experiments: Effect of Deep Memory

- Compare memory module (MLP) layer amount and parameter size
    - 170M, 360M, 760M
- Use Pile subset for training



Experiments: Time Series Forecasting

- Benchmarked on ETT, ECL, Traffic, and Weather datasets.
- Metric: **Mean Squared Error (lower is better)**.
- **Best model**: Neural Memory Module with **MSE = 0.162** (ECL dataset).
- Outperforms Simba, PatchTST, and TiDE.

	Neural Memory		Simba		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.358	0.387	0.383	0.396	0.407	0.410	0.414	0.407	0.387	0.400	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407
ETTm2	0.261	0.309	0.271	0.327	0.288	0.332	0.286	0.327	0.281	0.326	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401
ETTh1	0.420	0.421	0.441	0.432	0.454	0.447	0.446	0.434	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452
ETTh2	0.336	0.382	0.361	0.391	0.383	0.407	0.374	0.398	0.387	0.407	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515
ECL	0.162	0.261	0.169	0.274	0.178	0.270	0.219	0.298	0.205	0.290	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300
Traffic	0.415	0.289	0.493	0.291	0.428	0.282	0.626	0.378	0.481	0.304	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383
Weather	0.231	0.265	0.255	0.280	0.258	0.278	0.272	0.291	0.259	0.281	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317

Experiments (5/6): DNA Modeling

- Evaluated on GenomicsBenchmarks tasks.
- Metric: **Top-1 Classification Accuracy (higher is better)**.
- **Best model**: Neural Memory Module with **acc = 96.6%** (OCR task).
- Competitive with HyenaDNA and Transformer++.

Experiments (6/6): Efficiency & Scaling

- Compared training throughput and model scaling.
- **Best variant**: Titans (MAL) achieves **best trade-off** between speed and accuracy.
- **Scales up to 2M+ context window** with better efficiency than Mamba2 and DeltaNet.

Ablation Study

- Test Titans with and without certain parts
- Compare among three tasks with different

Model	Language Modeling ppl ↓	Reasoning acc ↑	Long Context acc ↑
LMM	27.01	47.83	92.68
+Attn (MAC)	26.67	48.65	97.95
+Attn (MAG)	25.70	48.60	96.70
+Attn (MAL)	25.91	47.87	96.91
Linear Memory	28.49	46.97	85.34
w/o Convolution	28.73	45.82	90.28
w/o Momentum	28.98	45.49	87.12
w/o Weight Decay	29.04	45.11	85.60
w/o Persistent Memory	27.63	46.35	92.49

Strengths & Unique Selling Points 💪

✅ Scales beyond 2M tokens 🚀
✅ Outperforms Transformers in recall-intensive tasks 🎯
✅ Surprise-based memory learning for better generalization 🔍
✅ Adaptive forgetting prevents memory overflow ⏳
✅ Parallelizable training for efficiency 🏎️

Criticisms & Limitations & Open Questions ❗

🚨 **High complexity** - Multiple interacting components 🤯
🚨 **Computational overheads** - Needs optimization for real-world use 🖥️
🚨 **Limited multimodal testing** - Lacks evaluation on vision + text 🌍
🚨 **Privacy concerns** - Memorization at test time could risk data leakage 🔐
🚨 **Reproducibility** - Many hyperparameters unknown until code is public
🚨 **Missing Information** - How large is the **persistent memory**?
🚨 **Experiment Design** - How often were the experiments repeated (no std. given)

Conclusion 🏆

✅ **Titans** introduce a scalable **long-term memory** framework 🔥
✅ Outperforms **Transformers and recurrent models** in long-context tasks 🎯
✅ Offers **better memory efficiency** with surprise-based learning 📚
✅ Future work: **Efficiency improvements, multimodal expansion**

Thank You! 🎉

🧑‍💻 Questions? Discussions?