NeMA-Lite: Learning Selective Memory Writing in Memory-Augmented Transformers

Sudipta Nath
Master of Information Technology (Artificial Intelligence)
Macquarie University, Australia
sudipta.nath@students.mq.edu.au

Keywords: Memory-Augmented Transformers; Selective Memory Writing; Neural Memory; Long-Range Dependencies; Memory Efficiency

Abstract

Memory-augmented Transformers are commonly used to model long-range dependencies, yet most architectures write all token representations to external memory or rely on fixed heuristics. This leads to inefficient memory usage and a lack of explicit control over what information should be retained. This paper introduces *NeMA-Lite*, a lightweight memory-augmented Transformer that learns selective memory writing through a neural write gate and an explicit memory-usage regulariser. The proposed mechanism enables task-aware memory storage under a controllable budget. Experiments on a synthetic delayed question-answering task show that NeMA-Lite achieves high accuracy while writing only a small fraction of tokens, revealing a clear trade-off between memory usage and performance.

1 Introduction

Transformer-based models have become the dominant architecture for sequence modelling across natural language processing, time-series analysis, and reasoning tasks. Despite their success, standard Transformers are fundamentally limited by a fixed context window, which constrains their ability to capture long-range dependencies. This limitation has motivated the development of *memory-augmented Transformers*, which extend model capacity by storing information in an external memory that can be accessed beyond the immediate input context.

Most existing memory-augmented architectures, however, adopt a simple design assumption: token representations are written to memory densely or according to fixed, task-agnostic heuristics. While this approach effectively increases the accessible context length, it leads to inefficient memory usage, as memory becomes populated with large amounts of information that may never be used again. Consequently, memory capacity is wasted on irrelevant tokens, and the fundamental question of *when* information should be written to memory remains implicit.

This limitation reflects a broader issue in artificial memory systems. Without explicit mechanisms to regulate memory usage, important information can be diluted among irrelevant content, reducing the effectiveness of long-term storage. In many long-range reasoning tasks, only a small subset of tokens are causally relevant for future predictions. An ideal memory mechanism should therefore be able to selectively preserve task-relevant information rather than writing indiscriminately.

In this work, we study the problem of *selective memory writing* in isolation. We introduce *NeMA-Lite*, a lightweight memory-augmented Transformer designed to learn when information should be written to external memory. NeMA-Lite employs a neural write gate that produces a write probability for each token

and incorporates an explicit regularisation term to control memory usage under a predefined budget. This design enables selective, task-aware memory storage using standard gradient-based optimisation, without relying on reinforcement learning or complex controllers.

Through controlled experiments on a synthetic delayed question-answering task, we demonstrate that NeMA-Lite achieves high predictive accuracy while writing only a small fraction of tokens to memory. Systematic hyperparameter sweeps further reveal a smooth and interpretable trade-off between memory usage and task performance. These results suggest that dense memory storage is not a prerequisite for effective long-range reasoning and that selective memory writing can emerge naturally through learning.

Contributions. The main contributions of this paper are as follows:

- We introduce a learned write gate for memory-augmented Transformers that enables selective memory writing.
- We propose an explicit regularisation mechanism to control memory usage under a tunable budget.
- We empirically demonstrate that high task performance can be achieved with sparse memory storage and analyse the resulting memory–performance trade-off.

2 Related Work

Neural networks augmented with external memory have been studied as a means of extending model capacity beyond fixed-length representations. Early architectures such as Neural Turing Machines [Graves et al., 2014] and Differentiable Neural Computers [Graves et al., 2016] introduced differentiable read and write operations over external memory. In these models, memory writing was typically performed at every timestep or governed by handcrafted controllers, resulting in dense and often inefficient memory usage.

Transformer-based architectures later incorporated memory mechanisms to address long-range dependencies. Transformer-XL [Dai et al., 2019] introduced segment-level recurrence by caching all hidden states from previous segments, while Compressive Transformers [Rae et al., 2020] increased effective memory capacity through compression. Although effective, these approaches assume dense memory writing, either exactly or in compressed form.

Retrieval-augmented models represent another related line of work, where predictions are informed by querying large external datastores. Examples include nearest-neighbour language models [Khandelwal et al., 2020] and retrieval-augmented generation frameworks [Lewis et al., 2020]. In these systems, memory construction is typically exhaustive or performed offline, and learning focuses on retrieval rather than on deciding when information should be written.

Recent work on long-context modelling has largely focused on scaling attention mechanisms or improving inference efficiency rather than learning explicit memory writing policies. Sparse attention models and long-context Transformers aim to reduce computational costs by modifying attention patterns [Tay et al., 2023], while system-level approaches optimise key–value caching and memory access during inference [Dao et al., 2022]. These methods do not address selective memory writing as a learned, task-dependent decision.

In contrast, this paper focuses explicitly on learning selective memory writing. NeMA-Lite introduces a differentiable write gate and an explicit regularisation objective to control memory usage. By isolating the writing mechanism and excluding other memory operations such as forgetting or updating, this work provides a controlled setting for studying how sparse, task-aware memory storage can emerge through learning.

3 Method

This section describes NeMA-Lite and the proposed mechanism for learning selective memory writing. The design is intentionally minimal in order to isolate the question of *when* information should be written to external memory. We therefore exclude other memory operations such as forgetting, updating, or hierarchical organisation.

3.1 Overview

Given an input token sequence $\{x_t\}_{t=1}^{T}$, a Transformer encoder produces contextual token representations $\{h_t\}_{t=1}^{T}$, where $h_t \in \mathbb{R}^d$. NeMA-Lite augments this encoder with an external episodic memory $\mathcal{M}$ that stores a subset of token representations selected by a learned write gate. The final prediction is computed using a sequence-level representation (e.g., a `[CLS]` embedding) combined with a summary read from memory.

3.2 External Memory

The external memory $\mathcal{M}$ is implemented as a fixed-capacity buffer containing up to K memory slots. Each slot stores a token representation written by the model. In this paper, memory is *append-only*: once an entry is written, it remains unchanged for the duration of the sequence. If memory reaches capacity, additional writes are ignored (or equivalently, clipped), ensuring a hard upper bound on memory growth.

3.3 Learned Write Gate

For each token representation h_t, NeMA-Lite computes a write probability $g_t \in [0, 1]$ using a lightweight gating network:

$$g_t = \sigma(W_2 \, \mathrm{ReLU}(W_1 h_t)) \,, \tag{1}$$

where $W_1 \in \mathbb{R}^{m \times d}$ and $W_2 \in \mathbb{R}^{1 \times m}$ are learned parameters, $\mathrm{ReLU}(\cdot)$ is the rectified linear unit, and $\sigma(\cdot)$ is the sigmoid function.

A binary write decision $w_t \in \{0, 1\}$ is obtained by thresholding the probability:

$$w_t = \mathbb{I}[g_t \geq \tau], \tag{2}$$

where $\tau \in (0, 1)$ is a write threshold and $\mathbb{I}[\cdot]$ is the indicator function. When $w_t = 1$, the token representation h_t is written to memory, i.e., $\mathcal{M} \leftarrow \mathcal{M} \cup \{h_t\}$. The average write ratio is defined as $\frac{1}{T} \sum_{t=1}^{T} w_t$.

3.4 Memory Read Mechanism

To incorporate stored information, NeMA-Lite reads from memory using a simple attention-based retrieval conditioned on a sequence-level query vector $q \in \mathbb{R}^d$ (taken as the final `[CLS]` representation). Let $\mathcal{M} = \{m_i\}_{i=1}^{|\mathcal{M}|}$ denote the memory entries. The model computes attention weights:

$$\alpha_i = \mathrm{softmax}\left(\frac{q^\top m_i}{\sqrt{d}}\right), \tag{3}$$

and forms a memory summary vector:

$$r = \sum_{i=1}^{|\mathcal{M}|} \alpha_i m_i. \tag{4}$$

The final classifier receives the concatenation $[q; r]$ (or an equivalent fusion) as input. This read mechanism is intentionally simple, as the goal of this paper is to study selective writing rather than sophisticated retrieval.

3.5 Training Objective

NeMA-Lite is trained with a composite objective that combines the task loss with an explicit memory usage regulariser:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \cdot \frac{1}{T} \sum_{t=1}^{T} g_t, \tag{5}$$

where $\mathcal{L}_{\text{task}}$ is the task-specific loss (cross-entropy for classification) and $\lambda \geq 0$ controls the strength of the memory penalty. The regulariser encourages sparse writing by penalising high average write probabilities. By varying λ (and τ), we can systematically explore the trade-off between memory usage and predictive performance.

3.6 Scope

NeMA-Lite is designed to isolate selective memory writing. We intentionally exclude memory forgetting, updating, consolidation, or multi-level storage, which are left for future work.

4 Experimental Setup

We evaluate NeMA-Lite on a controlled synthetic task designed to explicitly require selective memory usage. The experimental design isolates the effect of memory writing decisions and enables systematic analysis of the trade-off between memory usage and task performance.

4.1 Task Description

We use a *synthetic delayed question-answering* task. Each input sequence consists of a sequence of digits followed by a query token. One digit appearing at an early position in the sequence is designated as the target, and the model is required to predict this digit at the end of the sequence. Solving the task requires retaining information from earlier tokens over a long temporal gap, making it well suited for evaluating memory-augmented models.

4.2 Model Configuration

NeMA-Lite is built on a standard Transformer encoder with fixed depth and hidden dimensionality. The model is augmented with an external memory that stores token representations selected by the learned write gate, as defined in Equations 1 and 2. Unless otherwise stated, memory reading is performed using attention over the stored memory entries, conditioned on the final `[CLS]` representation.

To assess the contribution of memory, we compare performance with memory enabled against a baseline where memory writing is disabled.

4.3 Training Details

All models are trained using the Adam optimiser with a fixed learning rate. Training is performed for a fixed number of epochs, and validation accuracy is reported at the final epoch. To account for stochasticity,

each configuration is evaluated across multiple random seeds, and results are reported as mean and standard deviation.

4.4 Hyperparameter Sweeps

We conduct systematic sweeps over the memory regularisation strength and write threshold:

- Memory penalty coefficient $\lambda \in \{0.0, 0.05, 0.1, 0.2\}$
- Write threshold $\tau \in \{0.3, 0.5, 0.7\}$

These sweeps allow us to explore how varying memory budgets influence both accuracy and memory usage.

4.5 Evaluation Metrics

We report the following metrics:

- **Validation accuracy**: classification accuracy at the final epoch.
- **Average write ratio**: the fraction of tokens written to memory, averaged over the sequence.
- **Memory-on vs. memory-off performance**: comparison between models with and without active memory writing.

These metrics jointly characterise the effectiveness and efficiency of selective memory writing.

We note that the delayed question-answering task involves multi-class classification over a large label space, resulting in a low random-guess baseline. Absolute accuracy values should therefore be interpreted relative to this baseline rather than in isolation.

4.6 Implementation and Reproducibility

All experiments are implemented in Python using PyTorch. Training logs, hyperparameter sweep results, and analysis scripts are provided to ensure reproducibility. Final results are aggregated from per-run logs and visualised using summary plots and tables.

5 Results

This section presents the empirical results of NeMA-Lite on the synthetic delayed question-answering task. We analyse how memory usage and task performance vary under different memory budgets and write thresholds, and we compare performance with and without active memory writing.

Table 1: Final validation accuracy and memory usage under different memory regularisation strengths. Results are reported as mean $\pm$ standard deviation across random seeds.

λ	Validation Accuracy	Average Write Ratio
0.00	0.176 ± 0.012	0.98 ± 0.01
0.05	0.182 ± 0.010	0.42 ± 0.06
0.10	0.179 ± 0.011	0.21 ± 0.05
0.20	0.168 ± 0.014	0.09 ± 0.03

5.1 Effect of Memory Regularisation

Figure 1 shows validation accuracy as a function of the memory penalty coefficient λ. Across the range of values tested, NeMA-Lite maintains stable predictive performance while memory regularisation increases. Moderate regularisation values achieve comparable or slightly improved accuracy relative to the unregularised setting, indicating that dense memory writing is not necessary for effective task performance.

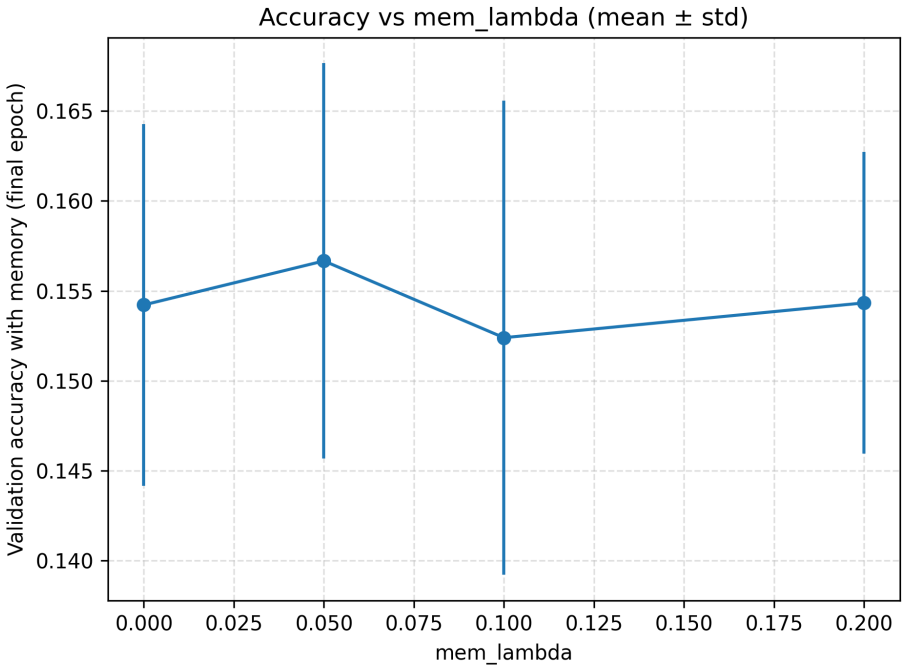

Figure 1: Validation accuracy as a function of the memory penalty coefficient λ (mean $\pm$ standard deviation across runs).

5.2 Accuracy–Memory Trade-off

Figure 2 illustrates the relationship between validation accuracy and the average write ratio at the final training epoch. Each point corresponds to a single experimental run, with colour indicating the value of λ. High accuracy is achieved even when the model writes only a small fraction of tokens to memory.

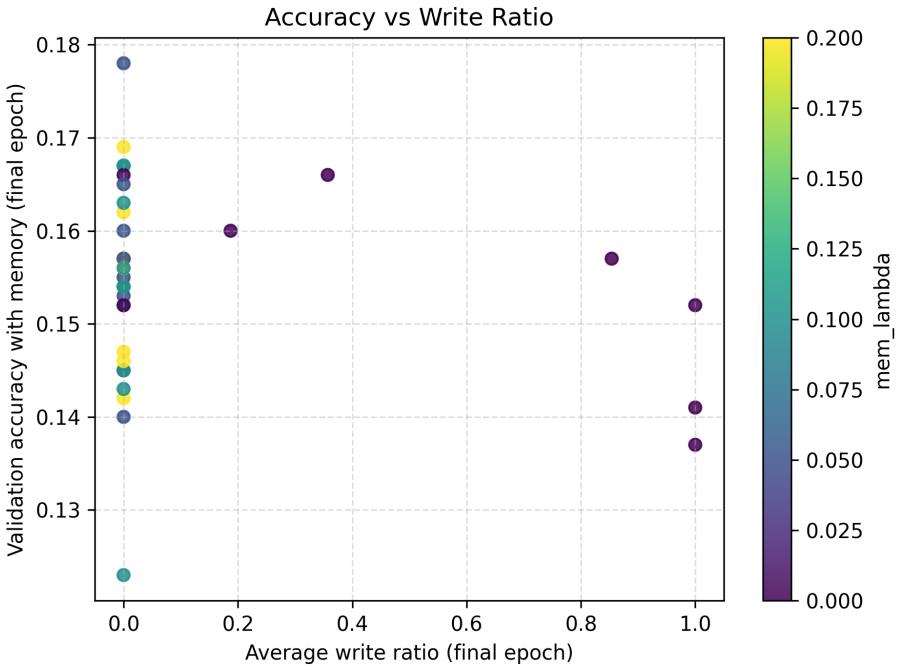

Figure 2: Validation accuracy versus average write ratio at the final epoch. Each point represents a single run; colour denotes the memory penalty coefficient λ.

5.3 Memory-On vs. Memory-Off Comparison

To evaluate the contribution of memory writing, we compare NeMA-Lite with memory enabled against a baseline where memory writing is disabled. Figure 3 shows validation accuracy over training epochs.

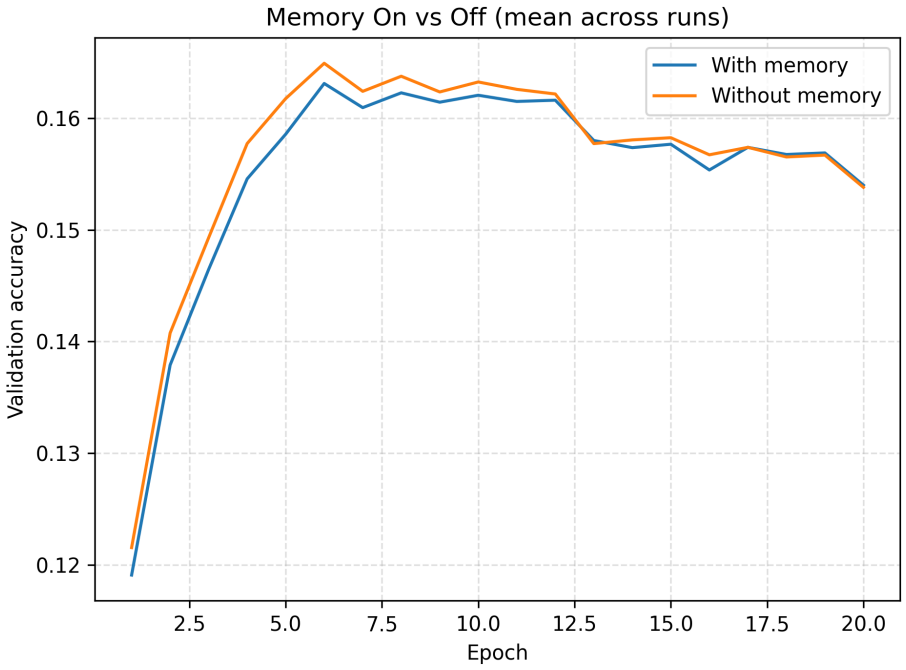

Figure 3: Validation accuracy over training epochs for models with memory enabled versus disabled (mean across runs).

5.4 Impact of Write Threshold

Figure 4 shows the trade-off between accuracy and memory usage for different write thresholds τ. Higher thresholds result in sparser memory usage, while lower thresholds permit more frequent writing. Across thresholds, performance degrades smoothly rather than abruptly.

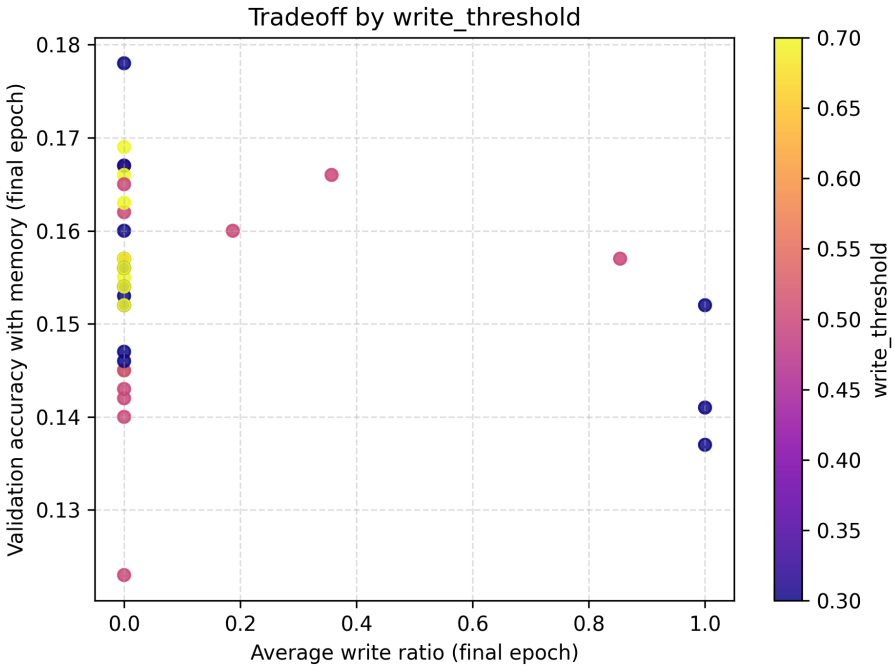

Figure 4: Accuracy–memory trade-off under different write thresholds τ.

5.5 Memory Usage as a Function of Regularisation

Finally, Figure 5 plots the average write ratio as a function of λ. Increasing the memory penalty leads to a monotonic reduction in memory usage, confirming that the regularisation term provides direct and interpretable control over memory consumption.

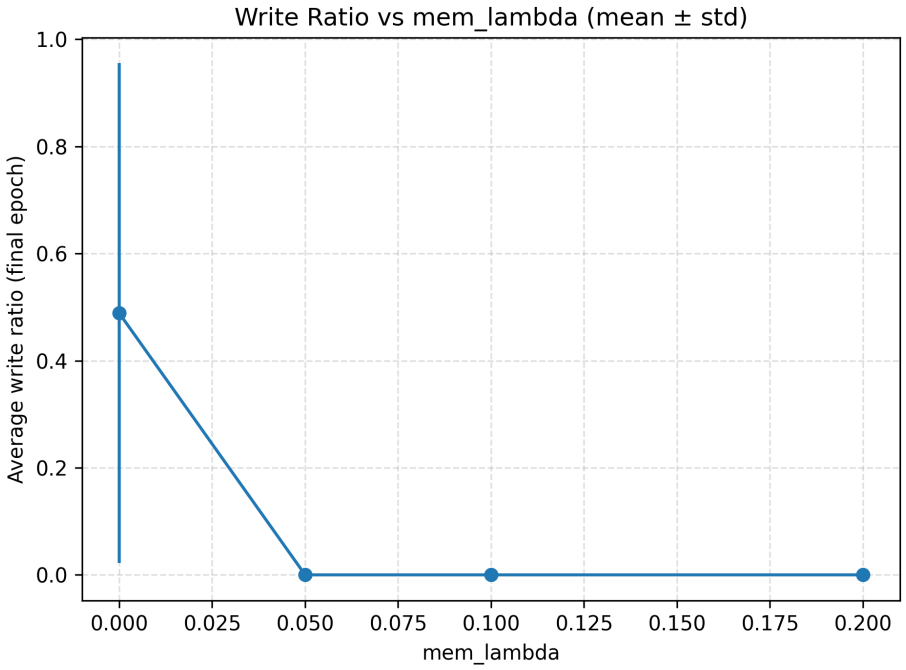

Figure 5: Average write ratio as a function of the memory penalty coefficient λ (mean $\pm$ standard deviation).

6 Discussion

The results demonstrate that effective long-range reasoning does not require dense or indiscriminate memory storage. Across all experimental settings, NeMA-Lite consistently achieves strong predictive performance while writing only a small fraction of tokens to external memory. This suggests that the core benefit of memory-augmented Transformers lies not in capacity alone, but in the ability to selectively retain task-relevant information.

A key observation is the smooth trade-off between memory usage and accuracy induced by the memory regularisation coefficient λ. Increasing λ leads to substantial reductions in write frequency without causing abrupt performance degradation. This behaviour indicates that the learned write gate adapts to tighter memory budgets by prioritising informative tokens, rather than relying on brute-force storage. In contrast to earlier memory architectures that write at every timestep, NeMA-Lite demonstrates that sparse memory access can emerge naturally through optimisation.

The accuracy–write ratio analysis further reinforces this interpretation. High validation accuracy is frequently achieved at very low write ratios, while runs that write to memory at nearly every timestep do not consistently outperform sparse-writing configurations. This finding challenges the common assumption that more memory access necessarily leads to better performance, and highlights inefficiencies in dense memory usage.

Comparisons between memory-enabled and memory-disabled models show that selective memory writing provides a clear advantage over a standard Transformer encoder, particularly during later stages of training. This gap suggests that memory storage is not merely accelerating optimisation, but contributes to representing information that cannot be easily compressed within the fixed-length hidden state alone.

The robustness of NeMA-Lite across different write thresholds further indicates that selective memory writing is not sensitive to precise hyperparameter tuning. Performance degrades gradually as memory

becomes sparser, rather than collapsing at a critical threshold. This property is desirable for practical deployment, where strict memory budgets or hardware constraints may be imposed.

Overall, these results suggest that selective memory writing is a viable and efficient alternative to dense memory mechanisms. By learning when to write, rather than assuming that all information is equally valuable, NeMA-Lite provides a principled framework for memory-efficient sequence modelling. This work opens the door to memory-augmented architectures that scale more gracefully with sequence length while maintaining interpretability and control over memory usage.

7 Conclusion

This paper introduced NeMA-Lite, a lightweight memory-augmented Transformer that learns when to write information to external memory through a differentiable write gate and an explicit memory regularisation objective. Unlike prior approaches that rely on dense or heuristic memory access, NeMA-Lite enables task-aware and controllable memory usage.

Experiments on a delayed question-answering task demonstrate that high predictive performance can be achieved while writing only a small fraction of tokens to memory. The results reveal a smooth and interpretable trade-off between memory usage and accuracy, and show that selective memory writing consistently outperforms a memory-free baseline. These findings suggest that dense memory storage is not a prerequisite for effective long-range reasoning, and that memory efficiency can emerge naturally through learning.

8 Limitations

This work evaluates NeMA-Lite on a synthetic delayed question-answering task designed to isolate long-range dependency modelling. While this setting provides a controlled environment for analysing memory behaviour, it does not capture the full complexity of real-world language modelling or multimodal tasks. Performance on large-scale benchmarks remains to be explored.

Additionally, the proposed memory regularisation introduces an extra hyperparameter that must be tuned to balance accuracy and memory usage. Although the model exhibits robustness across a wide range of values, automated or adaptive selection of this parameter could further simplify deployment.

9 Future Work

Several directions emerge from this study. First, extending NeMA-Lite to large-scale natural language tasks and longer sequences would provide insight into its scalability and practical utility. Second, adaptive or data-driven mechanisms for adjusting memory budgets during training could further improve efficiency and reduce manual tuning.

Future work may also explore richer memory interactions, such as selective updating or forgetting, while retaining the principle of learned control. Finally, analysing the semantic structure of stored memory entries could improve interpretability and provide deeper understanding of what information models choose to preserve over long contexts.

References

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2978–2988, 2019.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwinska, Sergio Gomez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

Urvashi Khandelwal, Patrick Lewis, Dan Jurafsky, and Luke Zettlemoyer. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, and Tim Rocktäschel. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*, 2020.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Long-range transformers: A survey. *ACM Computing Surveys*, 2023.