MSA: MEMORY SPARSE ATTENTION FOR EFFICIENT END-TO-END MEMORY MODEL SCALING TO 100M TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory is fundamental to intelligence, yet enabling AI to process lifetime-scale information on the order of hundreds of millions of tokens remains challenging. Full-attention architectures limit LLMs to roughly 1M tokens, and existing extensions such as linear attention, fixed-size memory, or external storage methods suffer from accuracy loss, growing latency, limited memory adaptability, or poor end-to-end optimization. These limitations restrict scalability and hinder potential applications such as large-corpus summarization, stable digital twins, and long-history agent reasoning. We present Memory Sparse Attention (MSA), an end-to-end trainable, efficient, and massively scalable memory model framework. Through core innovations including scalable sparse attention architecture and document-wise RoPE, MSA achieves linear complexity in both training and inference while maintaining exceptional precision stability, exhibiting less than 9% degradation when scaling from 16K to 100M tokens. Furthermore, KV cache compression, combined with Memory Parallel during inference, enables 100M tokens inference on $2 \times$ A800 GPUs. In addition, we propose a Memory Interleaving mechanism that effectively facilitates complex multi-hop reasoning across scattered memory segments. MSA significantly surpasses frontier language models, state-of-the-art (SOTA) RAG systems, and leading memory agents in long-context benchmarks. These results demonstrate that by decoupling memory capacity from reasoning, MSA provides a scalable foundation to endow general-purpose models with intrinsic, large-scale memory.

1 INTRODUCTION

While Large Language Models (LLMs) have demonstrated remarkable proficiency in competitive mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021), collaborative programming (Chen et al., 2021; Jimenez et al., 2024), and role-playing (Wang et al., 2024; Park et al., 2023), they remain confronted by a formidable challenge: long-term, fine-grained memory retention (Liu et al., 2024; Zhang et al., 2024b). Scenarios such as comprehending extensive novel series (Bai et al., 2024; Kočiský et al., 2018), maintaining consistent personas in role-playing, or managing the long-term history of multi-agent systems (Li et al., 2023; Park et al., 2023) place stringent demands on the model's memory capacity, specifically its effective context length. Research in cognitive science estimates the functional information capacity of human memory to be on the order of 10^9 bits (Landauer, 1986). Assuming an effective semantic density of 3–5 bits per token, this corresponds to a lifelong capacity of approximately 200–300 million tokens. This estimate motivates the capacity target of this work: approaching human-scale memory, as envisioned for potential applications such as digital twins, requires models that can handle contexts extending into the hundreds of millions of tokens. In stark contrast, contemporary LLMs typically support effective context lengths ranging from 128k to 1M tokens (Grattafiori et al., 2024; DeepSeek-AI et al., 2025; OpenAI et al., 2024). Even architectures explicitly designed for long contexts (Qwen et al., 2025; Yang et al., 2025), despite undergoing rigorous training pipelines, rarely exceed the 1M token threshold. To bridge this magnitude of disparity, a specialized mechanism for memory at this scale is needed.

An effective long-term memory system for LLMs should satisfy several core desiderata: seamless compatibility with mainstream model architectures, scalability to very large memory with low

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

computational overhead and minimal degradation in model quality, end-to-end trainable mechanisms that enable high-precision retrieval and storage, straightforward memory management, and robustness against catastrophic forgetting.

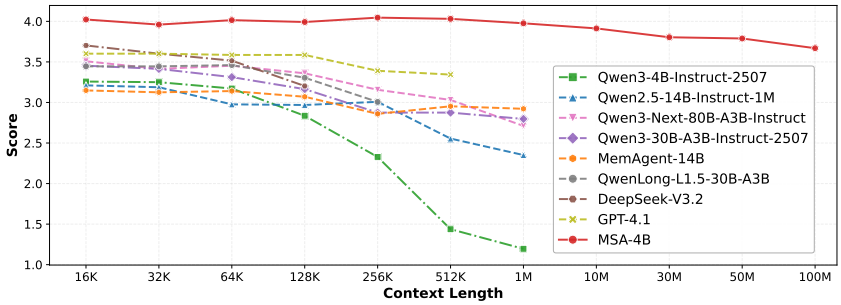

Figure 1: MSA integrates retrieval with sparse attention, achieving strong scalability while remaining end-to-end trainable. This design enables joint training of retrieval and generation, yet allows the documents to be decoupled at inference time, thereby providing robust extrapolation capability. MSA demonstrates exceptional scalability on the MS MARCO dataset, sustaining consistent performance with less than 9% degradation across an unprecedented memory context range from 16K to 100M tokens. Some curves terminate prematurely due to context length limitations.

As summarized in Table 1, current paradigms for LLM memory fall into three principal categories, each addressing only a subset of the essential criteria for scalable, high-fidelity long-term memory. **(I) Parameter-Based Memory** internalizes new knowledge by directly updating model parameters (e.g., LoRA (Hu et al., 2022), Continual Pre-training) or leveraging learnable architectures adapted via test-time training (e.g., Titans (Behrouz et al., 2025b)). While these methods provide strong architectural compatibility and high-precision semantic integration, they do not scale in capacity: model updates are prone to catastrophic forgetting, especially under conflicting knowledge, and require costly training with complex memory management. **(II) External Storage-Based Memory**, typified by Retrieval-Augmented Generation (RAG) and MemAgent, retrieves relevant information from large external knowledge stores. While this paradigm preserves base model capabilities and scales to very large memory banks without catastrophic forgetting, its reliance on an external retriever over discrete semantic representations prevents retrieval from being trained jointly with generation. The resulting decoupled retrieval pipeline caps performance, yielding only suboptimal precision and shallow semantic alignment. **(III) Latent State-Based Memory** aims to construct memory directly from internal latent representations (e.g., hidden states or KV caches), offering high semantic fidelity by operating within the model's native representation space. However, it faces a capacity–efficiency trade-off: KV-based methods such as DSA (DeepSeek-AI et al., 2025) and MemGen (Zhang et al., 2026) preserve precision and architectural compatibility but are prohibitively expensive to scale to 100M-token contexts, while linear-attention variants like RWKV (Peng et al., 2023) and DeltaNet (Yang et al., 2024b) achieve linear complexity via recurrent compression at the cost of bounded capacity, leading to catastrophic forgetting and degraded precision under extreme context lengths.

Overall, existing approaches remain constrained by two fundamental limitations: **(I) limited scalability of high-fidelity memory.** Methods that deliver strong precision are bound by fixed context or state capacity, while methods that scale in capacity struggle to ensure reliable effectiveness. **(II) lack of end-to-end trainability.** No current paradigm offers an end-to-end trainable, jointly optimized memory pipeline that simultaneously preserves architectural compatibility, high precision, and robustness against catastrophic forgetting across all scales. A comprehensive discussion of related work is provided in Appendix E.

To address these challenges, we propose Memory-Sparse Attention (MSA), a novel, end-to-end trainable, and scalable sparse attention mechanism designed specifically for large-scale memory contexts. As a latent state-based approach, MSA integrates top-k selection with sparse attention and trains retrieval and generation jointly within a single model. By leveraging KV cache sparsification, MSA achieves near-linear time complexity and supports inference over 100M tokens through optimized implementation. Furthermore, we introduce a global and document-wise Rotary Positional Embedding (RoPE) mixed strategy to extend the context window. This design allows MSA to be trained efficiently on 64k contexts while effectively extrapolating to 100M tokens, significantly

Table 1: Comparison of Long-Term Memory Methods for LLMs

Method	Lifetime Memory	Precision	Compatible w/ Mainstream LLMs	Computational Complexity	Memory Management	Catastrophic Forgetting
Parameter-Based Memory						
Model-Based (LoRA / CPT)	No	High	High	Training: High Inference: Low	Hard	Yes
Test-Time Training (Titans)	No	Medium	Low	Medium	Medium	Yes
External Storage-Based Memory						
RAG	Yes	Medium	Medium	$\mathcal{O}(L)$	Easy	Low
MemAgent	Yes	Medium	Medium	Medium	Easy	Medium
Latent State-Based Memory						
Sparse Attention (DSA)	No	High	High	Medium	Easy	No
Linear Attention (DeltaNet/RWKV)	No	Low	Low	$\mathcal{O}(L)$	Easy	Yes
MemGen	No	Medium	High	Medium	Medium	No
MSA (Ours)	**Yes**	**High**	**High**	$\mathcal{O}(L)$	**Easy**	**No**

reducing training overhead. Experimental results demonstrate that MSA achieves state-of-the-art (SOTA) performance on long-text Question Answering tasks, outperforming baseline models with identical backbones and surpassing advanced RAG systems on most benchmarks. Additionally, MSA achieves SOTA results on the "Needle-In-A-Haystack" (NIAH) test, exhibiting superior robustness against context degradation.

As illustrated in Figure 1, MSA demonstrates unprecedented scalability, maintaining performance with less than 9% degradation across context ranges spanning from 16K to 100 million tokens, a scale comparable to the estimated human-memory capacity that motivates this work. In comparison, traditional long-context models (e.g., Qwen2.5-14B-1M (Qwen et al., 2025), Qwen3-30B/80B-A3B (Yang et al., 2025)) and external memory systems (e.g., MemAgent-14B (Yu et al., 2026)) suffer from catastrophic degradation at this scale. Unlike SOTA RAG systems, MSA removes complex retrieval pipelines and heuristic tuning (e.g., top-k or relevance thresholds), narrowing the gap between LLM memory capacity and the human-scale target discussed above. Our contributions are summarized as follows:

- We propose MSA, an end-to-end trainable, scalable sparse attention architecture with a document-wise RoPE that extends intrinsic LLM memory while preserving representational alignment. It achieves near-linear inference cost and exhibits $< 9\%$ degradation even when scaling from 16K to 100M tokens.
- We introduce KV cache compression to reduce memory footprint and latency while maintaining retrieval fidelity at scale. Paired with Memory Parallel, it enables high-throughput processing for 100M tokens under practical deployment constraints, e.g., a single $2 \times$ A800 GPU node.
- We present Memory Interleave, an adaptive mechanism that facilitates complex multi-hop reasoning. By iteratively synchronizing and integrating KV cache across scattered context segments, MSA preserves cross-document dependencies and enables robust long-range evidence integration.
- Comprehensive evaluations on long-context QA and Needle-In-A-Haystack benchmarks demonstrate that MSA significantly outperforms frontier LLMs, state-of-the-art RAG systems and leading memory agents.

2 Memory Sparse Attention

2.1 Overall Design

We introduce Memory Sparse Attention (**MSA**), a unified, end-to-end trainable latent memory framework designed for massive memory Question-Answering. The core principle of MSA is to seamlessly integrate the processes of Memory sparse retrieval and answer generation into a single, jointly-optimized architecture, moving beyond the limitations of conventional decoupled "retrieve-then-read" pipelines while preserving the ability to handle long-context memory.

2.2 ARCHITECTURE

2.2.1 SPARSE ATTENTION MECHANISM

As shown in Figure 2, to efficiently process massive memory at the latent state level, MSA replaces the standard dense self-attention with a document-based retrieval sparse attention mechanism. Formally, let the memory bank consist of a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$. For each document d_i, let H_i denote its hidden state representation. For a specific attention head h, we generate the standard Key $K_{i,h}$ and Value $V_{i,h}$ matrices via the backbone model's projection weights W_K^h and W_V^h. In parallel, we introduce a ***Router K Projector***, parameterized by $W_{K^R}^h$, to generate a specialized routing key matrix $K_{i,h}^R$:

$$K_{i,h} = H_i W_K^h, \quad V_{i,h} = H_i W_V^h, \quad K_{i,h}^R = H_i W_{K^R}^h. \tag{1}$$

To significantly reduce the memory footprint and retrieval complexity, we segment each document into multiple fixed-length chunks and perform chunk-wise mean pooling, denoted as $\phi(\cdot)$, to compress these states into latent representations. This yields the compressed matrices $\bar{K}_{i,h} = \phi(K_{i,h})$, $\bar{V}_{i,h} = \phi(V_{i,h})$, and $\bar{K}_{i,h}^R = \phi(K_{i,h}^R)$.

During inference, given a user query with hidden state H_q, for a specific attention head, we similarly compute its standard states $Q_{q,h}, K_{q,h}, V_{q,h}$ via the backbone's W_Q^h, W_K^h, W_V^h projections. Simultaneously, a ***Router Q Projector*** $W_{Q^R}^h$ generates a specific routing query $Q_{q,h}^R = H_q W_{Q^R}^h$. The relevance score S_{ij} for the j-th chunk of the i-th document is computed as the cosine similarity between the query's routing vector $Q_{q,h}^R$ and the memory's compressed routing keys $\bar{K}_{ij,h}^R$, and is first aggregated across attention heads using mean pooling. To identify the most relevant memory segments, a maximum pooling is then applied over the query-token–level relevance scores, i.e.,

$$S_{ij}^{(\ell)} = \max_{\text{token } t} \big(\text{mean}_{\text{head } h}(\cos((Q_{q,h}^{R,(\ell)})_t, \bar{K}_{ij,h}^{R,(\ell)}))\big), \tag{2}$$

where ℓ indexes the MSA layer (omitted elsewhere) and $\cos(\cdot)$ denotes cosine similarity. The document-level relevance score is defined as the maximum score among its constituent chunks, $s_i = \max_j S_{ij}$. Based on these scores, we select the indices of the Top-k documents, denoted as $\mathcal{I} = \text{Top-}k(\{s_i\}_{i=1}^N)$. Finally, the generation is performed by concatenating the compressed Key and Value matrices of the selected documents before the query's local cache. The model then performs autoregressive generation where the query Q_q from active tokens attends to this aggregated, sparsity-aware context:

$$K_{\text{ctx}} = [\{\bar{K}_i\}_{i\in\mathcal{I}}; K_q], \quad V_{\text{ctx}} = [\{\bar{V}_i\}_{i\in\mathcal{I}}; V_q], \quad \text{Output} = \text{Attention}(Q_q, K_{\text{ctx}}, V_{\text{ctx}}). \tag{3}$$

Note that this layer-wise Top-k selection only determines which chunk-pooled, and hence lossy, compressed KV blocks enter the attention of that layer; it is not a retrieval of documents for reading. Original document text is fetched only through Generative Retrieval (Section 2.5), in which the model emits document IDs, so the number of documents read is decided by the model, not by k.

We apply MSA routing only to the upper half of the model, as early-layer representations lack the semantic abstraction needed for effective retrieval. In lower layers, we retain Independent Document Processing for state updates and alignment but omit sparse retrieval and memory integration, relying solely on local context without attending to compressed memory KV pairs.

By combining chunk-wise pooling with sparse top-k selection, MSA achieves $\mathcal{O}(L)$ complexity for both training and per-query inference with respect to the total memory size L. A detailed efficiency analysis is provided in Appendix C.

2.2.2 PARALLEL AND GLOBAL ROPE

To generalize across varying memory scales, MSA uses independent RoPE per document, addressing the train–test mismatch where models are trained on few documents but must infer over massive memory banks. Standard global positional encodings assign monotonically increasing position IDs across concatenated documents (Su et al., 2024), causing large position shifts as memory grows and degrading performance beyond the training context. By assigning independent position IDs (starting from 0) to each document, MSA decouples the positional semantics from the total number of documents in memory, enabling reliable extrapolation and maintaining high retrieval and reasoning accuracy on massive memory contexts despite training on smaller ones.

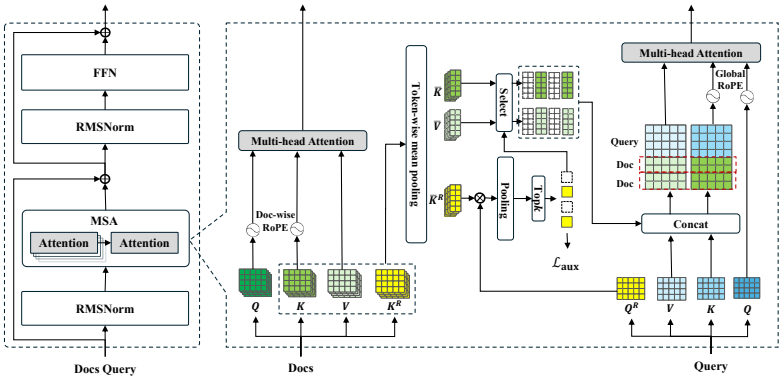

Figure 2: Memory Sparse Attention layer

Complementing the parallel strategy, MSA applies Global RoPE to the active context, including query and generation, with position IDs offset by the Top-k retrieved documents. Starting the query positions at k ensures the active context is treated as a continuation of the retrieved memory, preserving causal dependencies for coherent generation.

2.3 Training

2.3.1 Continual Pre-training

To endow the model with robust retrieval capabilities, we perform continual pre-training on a deduplicated corpus comprising 158.95 billion tokens. The overarching objective of this stage is to train the model to perform Generative Retrieval, where the model autoregressively generates the unique document IDs of relevant documents.

To explicitly guide the internal sparse attention mechanism beyond the supervision provided by the standard generation loss $\mathcal{L}_{\text{LLM}}$, we introduce an auxiliary loss, $\mathcal{L}_{\text{aux}}$, designed to supervise the Layer-wise Routing process. Within each MSA layer, the Router Projector is responsible for selecting the Top-k most relevant documents to participate in attention. We apply $\mathcal{L}_{\text{aux}}$ to these intermediate routing decisions to ensure that the model attends to the correct evidence. Inspired by Khosla et al. (2020), for a given query q, let $\mathcal{D}$ denote the associated document set, and let $\mathcal{P} \subseteq \mathcal{D}$ be the set of positive documents. The set of negative documents is $\mathcal{N} = \mathcal{D} \setminus \mathcal{P}$, whose cardinality is $|\mathcal{N}| = |\mathcal{D}| - |\mathcal{P}|$. Let s_i^+ denote the relevance score of the i-th positive query–document pair and $s_{i,j}^-$ the relevance score of the j-th negative paired with the i-th positive. The auxiliary loss is then defined as:

$$\mathcal{L}_{\text{aux}} = -\frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \log \frac{\exp\left(s_i^+/\tau\right)}{\exp\left(s_i^+/\tau\right) + \sum_{j=1}^{|\mathcal{N}|} \exp\left(s_{i,j}^-/\tau\right)}, \tag{4}$$

where τ is the temperature parameter. This supervised contrastive objective explicitly enforces separation between relevant and irrelevant document chunks in the latent routing space. The hard Top-k step is not differentiated through; the router is trained by $\mathcal{L}_{\text{aux}}$ (gradient path in Appendix I).

To ensure stability, we adopt a two-phase optimization schedule. In the initial warm-up phase, we focus on aligning these internal Router Projectors. We set the total loss to $\mathcal{L} = 0.1\mathcal{L}_{\text{LLM}} + \mathcal{L}_{\text{aux}}$ with a learning rate of 1e-4. This encourages the router heads to quickly learn effective selection policies. We then transition to the main pre-training phase, where the learning rate is annealed to 6e-6 and the loss weights are adjusted to $\mathcal{L} = \mathcal{L}_{\text{LLM}} + 0.1\mathcal{L}_{\text{aux}}$. This configuration prioritizes the ultimate Generative Retrieval task while maintaining the discriminative power of the internal layer-wise routing established during warm-up.

2.3.2 Post-Training

Following continual pre-training, we implement a two-stage curriculum learning strategy for SFT on Question Answering tasks. Firstly, we train on a large-scale dataset with an 8k context to establish core instruction-following and reasoning abilities within a standard context window. Secondly, we improve data quality and length generalization by filtering low-quality samples and extending the

context length to 64k, enabling robust adaptation to long dependencies and reliable extrapolation to massive memory banks during inference.

2.4 INFERENCE

The inference pipeline is designed to handle the large-scale memory bank efficiently through three distinct stages, as shown in Figure 3:

Stage 1: Global Memory Encoding (Offline). This stage performs a one-time offline pass over the corpus, where the model generates standard K and V matrices for each document, alongside routing keys K^R from a specialized *Router K Projector*. To reduce storage and retrieval costs, all matrices are chunked and mean-pooled into compact representations ($\bar{K}, \bar{V}, \bar{K}^R$), which are cached in the memory bank, transforming the raw corpus into a structured latent store. As documents are encoded independently, the bank can be edited per document by inserting or deleting its chunks (Appendix J).

Stage 2: Routing and Context Assembly (Online). Upon receiving a user query, the model projects its hidden states via the *Router Q Projector* to obtain a routing query Q_q^R, which is matched against cached routing keys $\bar{K}^R$ to select the Top-k relevant documents. For attention, only the compact Key–Value pairs ($\bar{K}, \bar{V}$) of these documents are loaded and concatenated with the query's local K_q, V_q, forming the final sparse context.

Stage 3: Sparse Generation (Online). In the final stage, the model operates autoregressively on the assembled sparse context. The standard attention mechanism computes the interaction between the active token's Query Q_q and the concatenated KV pairs $[\{\bar{K}_{topk}\}; K_q]$, generating the final answer token by token.

To enable inference over **100 million tokens** on a single 2×A800 node, we design a Memory Parallel pipeline that adopts tiered storage—keeping routing keys $\bar{K}^R$ on GPU while offloading content KVs ($\bar{K}, \bar{V}$) to CPU DRAM—and distributes the scoring computation across GPUs. Full details are provided in Appendix G.

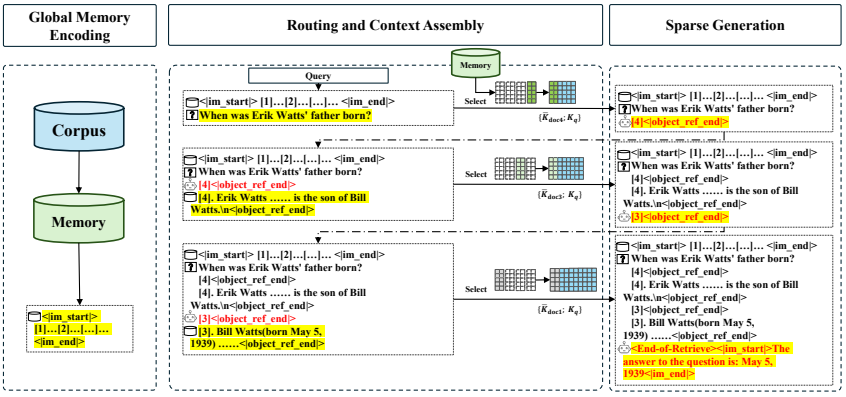

Figure 3: Three-Stage Inference Process with Memory Interleave

2.5 MEMORY INTERLEAVE

To address complex queries requiring multi-hop reasoning, MSA incorporates an adaptive Memory Interleave Mechanism that essentially performs the routing and context assembly (Stage 2) and Sparse Generation (Stage 3) in an iterative manner. Unlike single-shot retrieval, the inference process alternates between Generative Retrieval and Context Expansion, in which the retrieved documents will be treated as a part of the query for the next iteration, as shown in Figure 3. After loading the KV-cache for the document corpus, the model first autoregressively generates a sequence of document IDs ending with a special delimiter based on the given query. Note that the number of documents generated in each round is not fixed, but is adaptively determined by the model. Once the document IDs are generated, the system obtains the corresponding original texts and appends them to the original query, which is then leveraged in the next iteration. This cycle, which generates evidence identifiers, retrieves global context, and updates the state, repeats adaptively until the model

determines that the accumulated documents are sufficient, at which point it transitions from generating document IDs to autoregressively generating the final answer.

Notably, under the inference design described above, each retrieval chain in the multi-hop datasets is divided into multiple training samples during model training. Each sample contains a single retrieval step, either based on the single query or on the existing document context, and samples are randomly selected for training. Since at inference time later retrieval steps are conditioned on documents the model itself retrieved, we additionally augment these samples with cases whose first retrieval round contains incorrect documents, so that the model learns not to be derailed by wrong intermediate evidence.

3 EXPERIMENT

3.1 EXPERIMENTAL SETUP

We evaluate MSA on two tasks: (1) **Question Answering (QA)** over nine benchmarks with memory banks ranging from 277K to 10M tokens; and (2) **Needle-In-A-Haystack (NIAH)** using the RULER benchmark (Hsieh et al., 2024) with context lengths from 32K to 1M tokens.

For the QA task, we compare against two categories of baselines: (1) *same-backbone RAG*, and (2) *best-of-breed RAG* (KaLMv2-Embedding (Zhao et al., 2026) paired with Qwen3-235B or Llama-3.3-70B (Grattafiori et al., 2024) generators). We report performance using both an LLM-as-a-judge score on a 0–5 scale (prompt in Appendix A) and token-level F1 score. For RAG baselines, we present results at retrieval depth R@5 in this section, while more results are provided in Appendix H. For the NIAH task, we compare against MemAgent-14B (Yu et al., 2026) and hybrid linear attention models, including Qwen3-Next-80B-A3B, Qwen3-30B-A3B, and Qwen2.5-14B-1M. Full experimental details are in Appendix F.

3.2 MAIN RESULTS

3.2.1 QA TASK

Table 2: Comparison of MSA with same-backbone RAG baselines. "RR" denotes RAG systems with reranking.

	Tokens	Qwen3-4B		Qwen3-4B (RR)		HippoRAG2		MSA (Ours)	
Dataset		**LLM Score**	**F1**	**LLM Score**	**F1**	**LLM Score**	**F1**	**LLM Score**	**F1**
MS MARCO v1 (Bajaj et al., 2018)	7.34M	3.011	0.306	3.032	0.307	3.005	0.304	**4.141**	**0.687**
Natural Questions (Kwiatkowski et al., 2019)	1.47M	3.374	0.490	3.408	0.498	3.389	0.487	**3.545**	**0.534**
DuReader (He et al., 2018)	277K	3.579	0.482	3.618	0.484	3.485	0.278	**4.155**	**0.699**
TriviaQA (10M) (Joshi et al., 2017)	10M	4.414	0.718	4.375	0.728	4.430	0.732	**4.621**	**0.868**
NarrativeQA (Kočiský et al., 2018)	538K	2.567	0.330	**3.492**	0.451	2.628	0.249	3.395	**0.539**
PopQA (Mallen et al., 2023)	1.18M	3.273	0.266	3.264	0.271	3.249	**0.272**	**3.433**	0.228
2WikiMultiHopQA (Ho et al., 2020)	722K	3.055	0.542	3.057	0.540	3.180	0.569	**4.280**	**0.794**
HotpotQA (Yang et al., 2018)	1.35M	3.582	0.599	3.990	0.676	3.770	0.636	**4.061**	**0.717**
MuSiQue (Trivedi et al., 2022)	1.41M	1.752	0.257	1.960	0.290	1.907	0.272	**2.211**	**0.335**
Average		3.179	0.443	3.355	0.472	3.227	0.422	**3.760**	**0.600**

Table 3: Comparison of MSA with best-of-breed RAG baselines. "RR" denotes RAG systems with reranking.

	Qwen3-235B		Qwen3-235B (RR)		Llama3.3		Llama3.3 (RR)		MSA (Ours)	
Dataset	**LLM Score**	**F1**	**LLM Score**	**F1**	**LLM Score**	**F1**	**LLM Score**	**F1**	**LLM Score**	**F1**
MS MARCO v1	3.028	0.305	3.020	0.339	2.904	0.339	2.955	0.345	**4.141**	**0.687**
Natural Questions	3.670	0.541	3.610	0.548	**3.674**	**0.585**	3.665	0.579	3.545	0.534
DuReader	3.991	0.570	3.932	0.532	3.846	0.600	3.776	0.567	**4.155**	**0.699**
TriviaQA (10M)	4.656	0.771	4.320	0.737	**4.740**	0.818	4.703	0.839	4.621	**0.868**
NarrativeQA	2.130	0.272	**3.427**	0.451	2.123	0.281	3.263	0.449	3.395	**0.539**
PopQA	3.347	0.280	3.380	0.278	3.298	0.289	3.384	**0.291**	**3.433**	0.228
2WikiMultiHopQA	3.579	0.652	3.381	0.611	3.263	0.590	3.332	0.601	**4.280**	**0.794**
HotpotQA	4.090	0.708	4.141	0.714	3.896	0.667	**4.145**	0.709	4.061	**0.717**
MuSiQue	2.462	0.376	**2.522**	**0.385**	2.317	0.350	2.462	0.359	2.211	0.335
Average	3.439	0.497	3.526	0.511	3.340	0.502	3.521	0.527	**3.760**	**0.600**

On the QA benchmarks, MSA demonstrates consistent superiority over RAG baselines constructed with the identical Qwen3-4B-Instruct backbone, as shown in Table 2, achieving state-of-the-art performance on almost all datasets. MSA achieves substantial performance gains, yielding average LLM score improvements of 18.3%, 12.1%, and 16.5% and F1 improvements of 35.4%, 27.1%, and 42.2% over standard RAG, RAG with reranking, and HippoRAG2, respectively.

Against best-of-breed RAG systems, MSA secures top performance with the best average LLM score of 3.760 and F1 Score of 0.600. This represents average improvements of 8.8% on LLM score and 17.9% on F1 score over the four best-of-breed RAG baselines. A notable exception is PopQA, where MSA achieves the highest LLM Score (3.433) yet a lower F1 (0.228) than RAG baselines. This discrepancy arises because PopQA's gold answers are short entity aliases with special formatting, making token-level F1 highly sensitive to surface-form variations. All methods exhibit uniformly low F1 on this dataset (0.228–0.296), confirming that the gap reflects a metric artifact rather than a deficiency in answer quality.

3.2.2 NIAH TASK

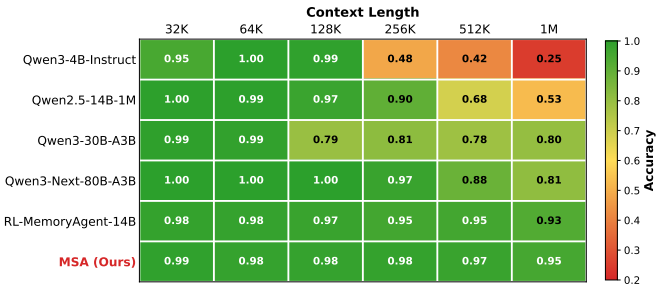

Figure 4: NIAH evaluation results across context lengths from 32k to 1M tokens.

On the NIAH benchmark, MSA remains stable from 32k to 1M tokens (Figure 4), decaying only gradually to 94.84% accuracy at 1M tokens. In contrast, the unmodified backbone (Qwen3-4B-Instruct) degrades sharply beyond 128k tokens, falling to 48.16% at 256k and 24.69% at 1M tokens.

Hybrid linear attention models also degrade under extreme scaling: Qwen2.5-14B-1M drops to 89.97% at 256k tokens, Qwen3-30B-A3B to 79.13% at 128k, and Qwen3-Next-80B-A3B, near-perfect up to 128k, to 80.78% at 1M tokens.

The external storage-based memory approaches MemAgent displays relatively stable performance without catastrophic failure points, yet its absolute accuracy remains consistently lower than MSA across all context lengths. More critically, its decay rate is markedly steeper: while MSA retains 94.84% accuracy at 1M tokens, reflecting only a 3.98% relative drop from its 32k-token performance of 98.77%, MemAgent declines to 92.66% at the same scale, a 5.85% relative drop. Overall, MSA is both more accurate and more robust at extreme context lengths than long-context architectures and external memory systems.

3.3 CONTEXT DEGRADATION

A persistent challenge in scaling language models is context degradation, where the accumulation of massive irrelevant context dilutes the model's ability to reason effectively. We evaluate MSA's robustness against this phenomenon on MS MARCO, measuring the LLM QA score as the memory context extends from 16K up to 100 million tokens. As illustrated in Figure 1, while state-of-the-art long-context models such as GPT-4.1 and DeepSeek-V3.2 (DeepSeek-AI et al., 2025) begin with competitive scores, they exhibit visible performance declines as the context length increases. In contrast, MSA demonstrates exceptional stability, starting with a strong score of 4.023 at 16K tokens and sustaining a competitive 3.669 even at the extreme 100M token scale. This represents a gradual degradation of only 8.8%. Conversely, the standard Qwen3-4B-Instruct backbone begins to degrade severely around 128k tokens and suffers a catastrophic collapse by 512k tokens. These results empirically validate that our sparse routing and independent positional encoding mechanisms effectively decouple reasoning capabilities from memory capacity, enabling MSA to operate reliably on massive-scale knowledge bases.

3.4 Ablation Study

Table 4: Ablation study on four QA benchmarks.

Model Variant	Average	MS MARCO v1	Natural Questions	DuReader	HotpotQA
MSA-S2 (Full)	**3.976**	**4.141**	**3.545**	**4.155**	**4.061**
MSA-S1 (Full)	3.694	3.197	3.493	4.064	4.020
w/o memory interleave	3.497	3.175	3.485	4.076	3.250
w/o continual pre-training	2.537	2.267	2.448	3.144	2.289
w/o original text	2.325	2.625	2.190	2.186	2.297

To systematically validate the contribution of each core component in MSA, we conduct comprehensive ablation experiments on four representative question answering benchmarks. As summarized in Table 4, we evaluate four critical design choices: (1) the impact of the two-stage curriculum learning strategy (MSA-S2 vs. MSA-S1), (2) the memory interleave mechanism for multi-hop reasoning (*w/o memory interleave*), (3) the continual pre-training stage (*w/o continual pre-training*), and (4) the integration of original document text after document ID generation (*w/o original text*).

Impact of Curriculum Learning. MSA-S2(full curriculum learning to 64k) reveals a 7.6% average performance gain against MSA-S1(post-trained at 8k context). This improvement is especially pronounced on datasets with massive memory banks: on MS MARCO (7.34M tokens), MSA-S2 achieves a remarkable gain of 29.5% over MSA-S1. These results provide compelling empirical validation that progressive exposure to longer contexts during training substantially enhances the model's ability to extrapolate to massive memory scales during inference.

Impact of Memory Interleave. The memory interleave mechanism delivers substantial gains on complex reasoning tasks. Removing this capability from the MSA-S1 baseline results in a 5.3% average performance degradation. The impact is particularly magnified on multi-hop datasets: HotpotQA experiences a significant 19.2% drop. This pattern confirms that memory interleave, where the model refines its retrieval query based on previously acquired evidence, is essential for compositional reasoning that requires evidence chains.

Impact of Continual Pre-training. CPT on large-scale retrieval tasks serves as the foundation for establishing robust routing capabilities. This stage incorporates a warmup phase that rapidly primes the router, significantly enhancing router-level precision, a factor pivotal to the model's overall retrieval efficacy. Eliminating CPT results in a severe average performance degradation of 31.3% (dropping from 3.694 to 2.537). This deterioration is consistent across all benchmarks, with the multi-hop dataset HotpotQA suffering a massive 43.1% decline. This sharp drop occurs because multi-hop tasks rely on memory interleaving, where errors in initial document retrieval accumulate during subsequent steps, thereby undermining the model's final reasoning performance.

Impact of Original Text. Integrating original document text after document ID generation provides substantial semantic grounding for answer synthesis. Disabling this component leads to the most severe average performance decline of 37.1% (from 3.694 to 2.325). Tasks requiring detailed reading comprehension suffer immensely: DuReader exhibits a massive 46.2% drop (4.064 to 2.186). This suggests that while document ID generation effectively localizes relevant evidence, the subsequent injection of raw document semantics remains essential for extracting the nuanced factual details necessary for precise response synthesis.

4 Conclusion

We introduce MSA, a scalable sparse-attention framework augmented with document-wise RoPE and KV-cache compression that extends end-to-end modeling to very large memory contexts, paired with Memory Parallel for fast 100M tokens processing and Memory Interleave for robust multi-hop reasoning across distributed memory segments. On long-context QA and Needle-in-a-Haystack benchmarks, MSA surpasses mainstream state-of-the-art general-purpose LLMs while preserving retrieval fidelity and reasoning depth. Crucially, performance degradation remains minimal even under extreme context lengths, maintaining high accuracy as the effective context scales to 100M tokens. These results indicate that by effectively decoupling memory capacity from reasoning capabilities, MSA can serve as a new foundational component to empower general-purpose models with memory capacity.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks for which the ICLR 2027 AI Policy requires disclosure. We used generative AI tools only to polish the writing of this paper, i.e., to edit the text for grammar and readability. All AI-edited text was reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes a model architecture and evaluates it on public QA and retrieval benchmarks; it involves no human subjects, user data, or deployment. Because it is motivated by persistent, large-scale memory, we note the implications of such a memory store. **What is retained.** Per document, the memory bank holds the chunk-pooled $\bar{K}$, $\bar{V}$ and routing keys $\bar{K}^R$, together with the source text, which is fetched for answer synthesis. Consent and retention policies must therefore govern the text store and not only the latent one; the compressed KV is a lossy but deterministic function of that text and should be treated as derived data, since compression is a storage property rather than a privacy guarantee. **Removal and correction.** A document can be removed exactly by dropping its chunks, without retraining and without residue in the weights (Appendix J), an advantage over parameter-based memory.

REPRODUCIBILITY STATEMENT

Model weights, code, pre-training data, and the evaluation benchmarks (including the 100M-token memory bank) are released in anonymous repositories: model `https://huggingface.co/Anoy123423123/MSA-4B`; code `https://huggingface.co/Anoy123423123/MSA-Code`; pre-training data `https://huggingface.co/datasets/Anoy123423123/MSA_PretrainData`; benchmarks `https://huggingface.co/datasets/Anoy123423123/MSA-RAG-BENCHMARKS`. Training data composition is given in Appendix B, all prompts in Appendix A, experimental settings and baselines in Appendix F, Memory Parallel inference details in Appendix G, and complete per-benchmark results in Appendix H.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, 2018. URL https://arxiv.org/abs/1611.09268.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=nbMeRvNb7A.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=8GjSf9Rh7Z.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gZyEJ2kMow.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. FireAct: Toward Language Agent Fine-tuning, 2023. URL https://arxiv.org/abs/2310.05915.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating Large Language Models Trained on Code, 2021. URL https://arxiv.org/abs/2107.03374.

Yuxuan Chen, Dewen Guo, Sen Mei, Xinze Li, Hao Chen, Yishan Li, Yixuan Wang, Chaoyue Tang, Ruobing Wang, Dingjun Wu, Yukun Yan, Zhenghao Liu, Shi Yu, Zhiyuan Liu, and Maosong Sun. UltraRAG: A Modular and Automated Toolkit for Adaptive Retrieval-Augmented Generation, 2025. URL https://arxiv.org/abs/2504.08761.

Xin Cheng, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Zhewen Hao, Han Zhang, Yu-Kun Li, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4968–4990, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.226. URL https://aclanthology.org/2026.acl-long.226/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. In *ECAI 2025*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2025. ISBN 9781643686318. doi: 10.3233/faia251160. URL http://dx.doi.org/10.3233/faia251160.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, et al. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models, 2025. URL https://arxiv.org/abs/2512.02556.

Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma GongQue, Weihao Zeng, Wei Wang, Jingang Wang, Xunliang Cai, and Weiran Xu. Agentrefine: Enhancing agent generalization through refinement tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FDimWzmcWn.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In Eunsol Choi, Minjoon Seo, Danqi Chen, Robin Jia, and Jonathan Berant (eds.), *Proceedings of the Workshop on Machine Reading for Question Answering*, pp. 37–46, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2605. URL https://aclanthology.org/W18-2605/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math

dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Thomas K. Landauer. How much do people remember? some estimates of the quantity of learned information in long-term memory. *Cognitive Science*, 10(4):477–493, 1986. ISSN 1551-6709. doi: 10.1207/s15516709cog1004_4. URL http://dx.doi.org/10.1207/s15516709cog1004_4.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In

H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023. doi: 10.52202/075280-2264. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, et al. GPT-4o System Card, 2024. URL https://arxiv.org/abs/2410.21276.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, pp. 1–22. ACM, 2023. doi: 10.1145/3586183.3606763. URL http://dx.doi.org/10.1145/3586183.3606763.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

Weijia Shi, Akshita Bhagia, Kevin Farhat, Niklas Muennighoff, Jacob Morrison, Evan Pete Walsh, Dustin Schwenk, Shayne Longpre, Jake Poznanski, Allyson Ettinger, Daogao Liu, Margaret Li, Mike Lewis, Wen tau Yih, Dirk Groeneveld, Luca Soldaini, Kyle Lo, Noah A. Smith, Luke Zettlemoyer, Pang Wei Koh, Hannaneh Hajishirzi, Ali Farhadi, and Sewon Min. FlexOLMo: Open language models for flexible data use. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1rUj9ZN6Bz.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL http://dx.doi.org/10.1016/j.neucom.2023.127063.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Noah Wang, Z.Y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.878. URL https://aclanthology.org/2024.findings-acl.878/.

Rubin Wei, Jiaqi Cao, Jiarui Wang, Jushi Kai, Qipeng Guo, Bowen Zhou, and Zhouhan Lin. MLP memory: A retriever-pretrained memory for large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1SMdxRtLBp.

Jing Xiong, Jianghan Shen, Chuanyang Zheng, Zhongwei Wan, Chenyang Zhao, Chiwun Yang, Fanghua Ye, Hongxia Yang, Lingpeng Kong, and Ngai Wong. ParallelComp: Parallel long-context compressor for length extrapolation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68998–69016. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/xiong25b.html.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=i2yIvZARnG.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, Chenyang Xi, Yu Yu, Kai Chen, Feiyu Xiong, Linpeng Tang, and Weinan E. Memory^3: Language modeling with explicit memory. *Journal of Machine Learning*, 3(3):300–346, 2024a. ISSN 2790-2048. doi: 10.4208/jml.240708. URL http://dx.doi.org/10.4208/jml.240708.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 115491–115522. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3668. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d13a3eae72366e61dfdc7eea82eeb685-Paper-Conference.pdf.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12380–12403, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.670. URL https://aclanthology.org/2024.acl-long.670/.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k5nIOvYGCL.

Guibin Zhang, Muxin Fu, and Shuicheng YAN. Memgen: Weaving generative latent memory for self-evolving agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vI56m4Iu4e.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Ming Zhu, Juntao Tan, Thai Hoang, Zuxin Liu, Liangwei Yang, Yihao Feng, Shirley Kokane, Tulika Awalgaonkar, Juan Carlos Niebles, Silvio Savarese, Shelby Heinecke, Huan Wang, and Caiming Xiong. AgentOhana: Design Unified Data and Training Pipeline for Effective Agent Learning, 2024a. URL https://arxiv.org/abs/2402.15506.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending long context evaluation beyond 100K tokens. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.814. URL https://aclanthology.org/2024.acl-long.814/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models, 2025. URL https://arxiv.org/abs/2506.05176.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, zhenyu liu, Dongfang Li, Xinyuan Wei, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. KaLM-embedding-v2: Superior training techniques and data inspire a versatile embedding model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Y7qzhvWhcz.

A PROMPTS

PROMPT TEMPLATE FOR LLM AS A JUDGE

```
Based on the accuracy, completeness, and relevance of the predicted
answer to the real answer in the context of the **query**, assign
an objective score from 0 to 5 (5 being the highest, 0 the lowest).

The scoring must strictly adhere to the following criteria.  The
final output can only be a single number.

Scoring Criteria:

5:  The predicted answer is exactly the same as the real answer and
correctly answers the query.  Differences in wording do not affect
factual accuracy.

4:  The predicted answer contains all the core information of
the real answer, with no errors, but includes a small amount of
non-critical redundant content.

3:  The predicted answer captures the core information but differs
from the real answer in some aspects.  The predicted answer is
slightly incomplete or imprecise, but contains no errors.

2:  The predicted answer is partially relevant to the real answer
but omits a significant amount of information or deviates from the
core topic of the query.

1:  The predicted answer attempts to address the query (maintains
basic relevance to the topic) but provides factually incorrect
information.  It does not contradict the core claim of the real
answer, but shows incomplete or inaccurate understanding of the
topic.

0.  The predicted answer is completely unrelated to the query,
consists of gibberish, or is a pure hallucination that shares no
logical connection with the real answer.

Query:

{query}

True Answer:

{gold_answer}

Predicted Answer:

{model_answer}

Output only a single number (0, 1, 2, 3, 4, or 5):
```

Table 5: Detailed statistics of the full MSA Pre-training Dataset.

Dataset Source (Filename)	Queries	Tokens	Task / Domain
Long-Context & Instruction Tuning			
kalmfinetune_data	5,801,540	6.46B	Knowledge Augmentation
Academic & Scientific Literature			
S2ORC_citations_abstracts	500,000	7.31B	Scientific Literature
S2ORC_citations_titles	500,000	7.18B	Scientific Literature
S2ORC_title_abstract	500,000	7.11B	Scientific Literature
specter_train_triples	500,000	7.14B	Scientific Citation
General QA & Community Knowledge			
yahoo_answers_qa	500,000	7.10B	General QA
yahoo_answers_ta	500,000	7.09B	General QA
yahoo_answers_tq	500,000	7.10B	General QA
WikiAnswers	500,000	7.16B	Community QA
gooaq_pairs	500,000	7.11B	FAQ / Common QA
msmarco_triples	499,184	7.37B	Information Retrieval
PAQ_pairs	500,000	7.15B	Synthetic QA
amazon_qa	500,000	7.11B	E-commerce QA
eli5_question_answer	325,475	4.62B	Explainable QA
stackexchange_body_body	250,460	3.58B	Technical QA
stackexchange_title_body	250,519	3.59B	Technical QA
stackexchange_title_title	304,525	4.31B	Technical QA
searchQA_top5_snippets	117,220	1.68B	Machine Reading Comprehension
quora_duplicates	103,663	1.46B	Duplicate Detection
quora_duplicates_triplets	101,762	1.45B	Duplicate Detection
NQ_train_pairs	100,231	1.43B	Open-Domain QA
squad_pairs	87,599	1.24B	Reading Comprehension
TriviaQA_pairs	73,346	1.04B	Trivia QA
News & Summarization			
agnews	500,000	7.10B	News Classification
npr	500,000	7.01B	News Broadcast
ccnews_title_text	500,000	6.96B	News Data
cnn_dailymail_splited	311,971	4.70B	Long-Document Summarization
cnn_dailymail	311,971	4.41B	News Summarization
xsum	226,711	3.20B	Extreme Summarization
sentence_compression	180,000	2.55B	Text Compression
altlex	112,696	1.60B	Paraphrasing
Domain-Specific & Others			
amazon_review_2018	500,000	7.08B	E-commerce Reviews
codesearchnet	500,000	7.01B	Code / Programming
AIINLI	277,230	3.94B	Natural Language Inference
wikihow	128,543	1.82B	Instructional / How-to
SimpleWiki	102,225	1.46B	General Knowledge
coco_captions	82,783	1.18B	Image Captioning
flickr30k_captions	31,783	0.45B	Image Captioning
Total	**17,852,825**	**158.95B**	**All Domains**

B Pre-training Data Composition

To ensure the model possesses both robust retrieval capabilities and broad general knowledge, we constructed a diverse pre-training corpus comprising 158.95 billion tokens across 17.9 million queries. As detailed in Table 5, the corpus covers a wide range of domains from scientific literature to general community Q&A. To maintain a balanced data distribution, we downsample any dataset outside the KALM suite that exceeds 0.5 million queries to a maximum of 0.5 million, while retaining the KALM instruction data in its entirety.

C Efficiency Analysis

We analyze the computational complexity of Memory Sparse Attention with respect to memory size L. We define M as the query length ($M \ll L$), G as the average document length, k as the number

of top-k documents selected (a small constant, e.g., 16), and P as the chunk-wise pooling size (fixed at 64 in practice). MSA achieves linear complexity with respect to L in both training and inference regimes, as detailed below.

C.1 Training Complexity

During training, MSA processes the entire memory bank within each forward pass. The computational cost consists of three components:

1. **Independent Document Processing**: Each of the L/G documents undergoes intra-document self-attention independently. With $\mathcal{O}(G^2)$ complexity per document, the aggregated cost across all documents is $\mathcal{O}(LG)$.
2. **Sparse Routing**: The model computes relevance scores between the query representation and the L/P pooled chunks from the memory bank, incurring $\mathcal{O}(ML/P)$ complexity.
3. **Sparse Generation**: Attention is applied over the concatenated context comprising the query and the k selected documents (each compressed to G/P chunks). This stage costs $\mathcal{O}\big((M + kG/P)^2\big)$, which depends only on query length and fixed hyperparameters (k, P), and is therefore independent of the total memory size L.

Summing these components yields the total training complexity:

$$\mathcal{O}_{\text{train}} = \mathcal{O}(LG) + \mathcal{O}(ML/P) + \mathcal{O}\big((M + kG/P)^2\big) = \mathcal{O}(LG) \tag{5}$$

where the $\mathcal{O}(LG)$ term dominates when $L \gg G$, which holds in extreme-long memory scenarios.

C.2 Inference Complexity

MSA's inference pipeline separates computation into offline pre-processing and online query handling:

1. **Offline Pre-processing**: Prior to serving queries, the system performs a one-time forward pass over the entire document collection to generate and cache compressed representations ($\bar{K}$, $\bar{V}$, $\bar{K}^R$) for all documents. This stage incurs $\mathcal{O}(LG)$ complexity but is executed only once per memory bank version—unlike conventional attention mechanisms that require $\mathcal{O}(L^2)$ prefill computation for every query.
2. **Online Routing**: For each incoming query, the model matches the routing query against the precomputed cache of L/P entries, costing $\mathcal{O}(ML/P)$.
3. **Online Generation**: Autoregressive generation operates over the assembled sparse context of size $M + kG/P$. With T denoting the answer length, this stage costs $\mathcal{O}\big(T \cdot (M + kG/P)^2\big)$, which remains independent of L.

The per-query inference complexity is therefore:

$$\mathcal{O}_{\text{inference}} = \mathcal{O}(ML/P) + \mathcal{O}\big(T \cdot (M + kG/P)^2\big) = \mathcal{O}(L) \tag{6}$$

where the routing term $\mathcal{O}(ML/P)$ dominates and scales linearly with memory size. Crucially, the expensive $\mathcal{O}(LG)$ pre-processing is amortized across all queries served from the same memory bank, yielding substantial efficiency gains in query-heavy workloads compared to methods requiring per-query $\mathcal{O}(L^2)$ prefill operations.

D Limitations

Although this work enhances intrinsic latent-state memory for long textual contexts, it remains limited when tasks require modeling strong and tightly coupled dependencies across multiple documents. In scenarios where evidence is distributed and highly interlinked across sources, the method struggles to maintain accurate structural alignment purely through intrinsic memory. Memory interleave is a potentially promising direction for mitigating these issues, as it can help integrate and synchronize information from separated context segments. However, its effectiveness depends on more efficient and principled designs that better preserve inter-document relationships.

E Related Work

As outlined in the introduction, recent research on augmenting LLMs with memory capabilities generally falls into three paradigms.

Parameter-based memory. This paradigm seeks to internalize external information directly into the model's parameters. A foundational approach involves direct fine-tuning on domain-specific data using techniques such as Continual Pre-training (CPT) or LoRA. This strategy is widely adopted to embed procedural knowledge and reasoning patterns (Chen et al., 2023; Yin et al., 2024; Zhang et al., 2024a; Fu et al., 2025). To mitigate catastrophic forgetting and decouple memory from reasoning, recent research has shifted towards specialized architectural components. MLP-Memory (Wei et al., 2026), for instance, substitutes explicit retrieval with a parametric retriever, training an MLP to act as a differentiable memory store. Scaling this modular concept further, FLEXOLMO (Shi et al., 2025) introduces a mixture-of-experts framework that updates specific modules for targeted knowledge integration, while Engram (Cheng et al., 2026) augments the model with massive sparse memory structures via N-gram embeddings to bypass the capacity bottlenecks of dense layers. Pushing the paradigm towards "dynamic neural memory," recent innovations such as Titans (Behrouz et al., 2025b) and Nested Learning (Behrouz et al., 2025a) propose maintaining memory modules whose weights are updated during inference (test-time training), treating context processing as a nested optimization loop. This direction is theoretically grounded in frameworks like MIRAS (Behrouz et al., 2026), which unifies such recurrent and associative memory architectures under a common abstraction.

External storage-based memory. This paradigm augments models with a large-scale external database, from which relevant memories are extracted via semantic retrieval on demand. The foundational framework in this category is Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), which retrieves textual chunks based on vector similarity between the query and the external corpus. To address the precision limitations of initial dense retrieval, which can introduce irrelevant or "noisy" context, state-of-the-art RAG systems frequently incorporate a reranking stage to refine the candidate list, ensuring that only the most pertinent information occupies the model's limited context window. Recent innovations have sought to optimize the format of retrieved memory. Memory^3 (Yang et al., 2024a), for instance, pre-encodes external knowledge into structured KV pairs for direct injection into the model's attention layers. Crucially, however, the retrieval process in Memory^3 remains grounded in model-agnostic semantic embeddings rather than the model's internal state, maintaining an optimization gap between the retrieval metric and the generation objective. To bridge this gap, MemAgent (Yu et al., 2026) formulates memory management as a sequential decision-making process. By employing Reinforcement Learning, it trains the model to actively read, write, and overwrite memory segments, thereby aligning the information retention policy directly with the downstream reasoning performance rather than relying solely on static similarity metrics. Addressing the structure of memory, MemGAS (Xu et al., 2026) improves upon the flat indexing of standard RAG by introducing a hierarchical management mechanism. This allows for multi-granularity retrieval, enabling the system to adaptively fetch information ranging from coarse-grained summaries to fine-grained details depending on the specific query requirements.

Latent state-based memory. Distinct from model-agnostic semantic retrieval-based memory, the latent memory paradigm constructs and manages memory directly using the model's internal latent states. As noted previously, Memory^3 attempts to leverage this by encoding information into KV pairs; however, constrained by the prohibitively large size of active KV caches, it offloads these representations to an external database. Consequently, it still relies on model-agnostic semantic embeddings as retrieval keys to concatenate retrieved pairs with the context, rather than maintaining a persistent internal state. In contrast, more intrinsic approaches aim to manage the model's working memory directly. ParallelComp (Xiong et al., 2025) addresses the capacity limit by implementing sophisticated KV cache eviction policies to dynamically compress context during inference. Similarly, MemGen (Zhang et al., 2026) exploits the model's autoregressive capabilities to iteratively synthesize and compress historical information into compact memory representations, thereby retaining essential information within the model's latent space.

Another distinct class of latent memory is Linear Attention mechanisms. In contrast to standard attention, which requires explicit access to previous KV, linear attention naturally compresses information from the preceding sequence into compact hidden states during the recurrence. Architectures such as

RWKV (Peng et al., 2023) formulate attention as a linear recurrence (WKV), where historical context is aggregated into a time-decaying hidden state. Similarly, DeltaNet (Schlag et al., 2021; Yang et al., 2024b) updates its memory state using a delta rule, iteratively refining value representations based on new inputs. While compressing the entire history into fixed-size latent states yields substantial computational and storage efficiency, it inherently involves lossy compression. Consequently, when constrained by a finite state size, these methods inevitably suffer from severe performance degradation and information loss as the memory context extends to extreme-long scales.

F Experimental Details

Datasets and Metrics. We evaluate MSA on nine diverse benchmarks covering single-hop, multi-hop, and long-context scenarios: MS MARCO v1 (Bajaj et al., 2018) (7.34M tokens), Natural Questions (Kwiatkowski et al., 2019) (1.47M), DuReader (He et al., 2018) (277K), TriviaQA (10M) (Joshi et al., 2017), NarrativeQA (Kočiský et al., 2018) (538K), PopQA (Mallen et al., 2023) (1.18M), 2WikiMultiHopQA (Ho et al., 2020) (722K), HotpotQA (Yang et al., 2018) (1.35M), and MuSiQue (Trivedi et al., 2022) (1.41M). For the standard RAG systems, we report performance metrics (LLM judge, whose prompt is shown in Appendix A and token-level F1 Score; the LLM judge is Gemini-2.5-Flash with temperature 0, accessed via OpenRouter, and is used for all systems) at fixed retrieval depths of $k = \{1, 5, 10\}$, denoted as **R@1, R@5, and R@10**, respectively. Notably, for the RAG systems that perform retrieval and reranking, we first retrieve 100 candidate documents and then select the top-$\{1, 5, 10\}$ items based on the reranked list. In contrast, for our MSA models, we utilize an **@adaptive** metric. This indicates that instead of relying on a pre-defined, fixed number of retrieved documents, the model autonomously determines the number of documents required to answer each specific query. For NIAH evaluation, we employ the RULER benchmark (Hsieh et al., 2024), which consists of eight diverse sub-tasks covering both standard single-needle retrieval (SA1-3) and complex multi-needle scenarios involving multiple keys, values, and queries (MK1-3, MV, MQ). We report the average accuracy across these tasks to comprehensively assess the model's stability and extrapolation capabilities from 32K up to 1M tokens.

Implementation Details. Our MSA model is built upon the Qwen3-4B-Instruct-2507 architecture. We initialize the backbone parameters using the official pre-trained weights to leverage its established capabilities, while the newly introduced router projectors are randomly initialized. The model undergoes continual pre-training on the 158.95B token corpus. To ensure stability, we employ the two-stage pre-training schedule described in Sec. 2.3, transitioning from a retrieval-focused warmup ($\mathcal{L} = 0.1\mathcal{L}_{\text{LLM}} + \mathcal{L}_{\text{aux}}$) to the main pre-training phase ($\mathcal{L} = \mathcal{L}_{\text{LLM}} + 0.1\mathcal{L}_{\text{aux}}$). Regarding the specific MSA hyperparameters, we set the compression chunk size to 64 tokens and configure the router to select the Top-16 relevant documents for attention. We evaluate two model variants to analyze the impact of our curriculum learning strategy: MSA-S1, which is fine-tuned solely through the first stage of post-training with a standard 8k context; and MSA-S2, which undergoes the complete two-stage curriculum learning pipeline, extending the memory context to 64k.

Baselines. We evaluate MSA on QA task and NIAH task with task-specific baselines for each. For QA task, we compare against two categories of RAG baselines. (I) same-backbone RAG: MSA is initialized with Qwen3-4B-Instruct-2507 (Yang et al., 2025). To validate the effectiveness of MSA, we evaluate RAG systems built on the same backbone, including standard RAG (Qwen3-4B-Embedding (Zhang et al., 2025) + Qwen3-4B-Instruct-2507), RAG with reranking (adding Qwen3-4B-Rerank), and HippoRAG2 (Gutiérrez et al., 2025), a knowledge graph-augmented RAG framework. (II) best-of-breed RAG: We further compare against state-of-the-art configurations employing KaLMv2-Embedding-Gemma3-12B-2511 (Zhao et al., 2026) as the retriever, paired with frontier-scale generators including Qwen3-235B-Instruct-2507 and Llama-3.3-70B-Instruct (Grattafiori et al., 2024), with optional reranker Qwen3-8B-Rerank. Unless otherwise noted, all standard RAG pipelines are implemented using UltraRAG v2.0 (Chen et al., 2025). For NIAH task, we compare against external storage-based memory method MemAgent-14B (Yu et al., 2026) and mixed linear attention models, including Qwen3-Next-80B-A3B, Qwen3-30B-A3B, Qwen2.5-14B-1M. Additionally, our backbone model Qwen3-4B-Instruct-2507 is also included.

G Memory Parallel Details

Operating on a 100M-token memory context presents significant challenges in terms of memory capacity and computational efficiency. To address these, we implement a tailored optimization pipeline, Memory Parallel, covering the entire lifecycle from encoding to retrieval.

Tiered Memory Storage Strategy. For the runtime storage, a theoretical estimation indicates that the compressed KV and $\bar{K}^R$ cache for 100M tokens, assuming a pooling kernel size $P = 64$, 8 heads, and head dimension 128 across 18 layers in BF16, would require approximately 169GB of memory. This figure strictly exceeds the aggregate 160GB capacity of a standard 2×A800 node, rendering a monolithic storage approach physically impossible, even before accounting for the memory required by model parameters and dynamic activation overheads. We observe that retrieval only requires the routing keys $\bar{K}^R$, while the content $\bar{K}$ and $\bar{V}$ are needed only after selection. Thus, we design a tiered storage system:

- **GPU-Resident Routing Keys:** To ensure low-latency retrieval, we distribute the Routing Keys ($\bar{K}^R$) across the VRAM of multiple GPUs. Even with optimizations, $\bar{K}^R$ alone can occupy ∼56GB for a 100M context, necessitating distributed storage.
- **CPU-Offloaded Content KVs:** The bulk of the memory bank, the Content KVs ($\bar{K}, \bar{V}$), is stored in the host DRAM (CPU memory). Upon identifying the Top-k relevant chunks via GPU scoring, only the corresponding Content KVs are asynchronously fetched from the host to the GPU for the subsequent attention computation.

This separation decouples the capacity requirement from VRAM limits, enabling 100M-token scale on standard hardware.

Memory-Parallel Retrieval and Distributed Scoring. To address computational efficiency, we adopt a Memory Parallel strategy for retrieval. Given the relatively compact size of the 4B backbone, we replicate the full model weights on each GPU to avoid communication overhead during decoding. During the retrieval step, the query hidden states are broadcast to all GPUs. Each GPU independently calculates similarity scores against its local shard of Routing Keys. These scores are then reduced globally to identify the Top-k indices. Additionally, we implement a tiling mechanism for the scoring matrix multiplication to manage peak memory usage and prevent OOM errors during these large-scale operations.

H Full QA Benchmark Results

Tables 6 and 7 present the complete LLM judge scores at all retrieval depths (R@1, R@5, R@10) for same-backbone and SOTA RAG baselines, respectively, corresponding to the condensed Tables 2 and 3 in the main text. Tables 8 and 9 present the corresponding complete F1 results at all retrieval depths.

Table 6: Full results: Comparison of MSA with same-backbone RAG baselines on LLM judge results (scale 0–5) at all retrieval depths R@{1,5,10}. "RR" denotes RAG systems that perform both retrieval and reranking.

	Tokens	Qwen3-4B			Qwen3-4B (RR)			HippoRAG2			MSA (Ours)
Dataset		**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**@adaptive**
MS MARCO v1 (Bajaj et al., 2018)	7.34M	2.893	3.011	3.005	2.934	3.032	3.017	2.676	3.005	3.019	**4.141**
Natural Questions (Kwiatkowski et al., 2019)	1.47M	3.452	3.374	3.297	3.494	3.408	3.385	3.338	3.389	3.374	**3.545**
DuReader (He et al., 2018)	277K	3.726	3.579	3.594	3.848	3.618	3.607	2.941	3.485	3.415	**4.155**
TriviaQA (10M) (Joshi et al., 2017)	10M	4.133	4.414	4.273	4.313	4.375	4.391	4.188	4.430	4.367	**4.621**
NarrativeQA (Kočiský et al., 2018)	538K	1.611	2.567	2.860	**3.638**	3.492	3.536	1.959	2.628	2.655	3.395
PopQA (Mallen et al., 2023)	1.18M	2.959	3.273	3.299	3.315	3.264	3.266	3.111	3.249	3.249	**3.433**
2WikiMultiHopQA (Ho et al., 2020)	722K	1.065	3.055	3.136	1.187	3.057	3.159	1.045	3.180	3.330	**4.280**
HotpotQA (Yang et al., 2018)	1.35M	2.252	3.582	3.787	2.642	3.990	4.022	3.230	3.770	3.970	**4.061**
MuSiQue (Trivedi et al., 2022)	1.41M	0.936	1.752	1.928	1.144	1.960	1.965	1.020	1.907	2.095	**2.211**
Average		2.559	3.179	3.242	2.946	3.355	3.372	2.612	3.227	3.275	**3.760**

Table 7: Full results: Comparison of MSA with SOTA RAG systems using large-scale backbones on LLM judge results (scale 0-5) at all retrieval depths R@{1,5,10}. "RR" denotes RAG systems that perform both retrieval and reranking.

	Qwen3-235B			Qwen3-235B (RR)			Llama3.3			Llama3.3 (RR)			MSA (Ours)
Dataset	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**@adaptive**
MS MARCO v1	2.846	3.028	3.027	2.886	3.020	2.995	2.649	2.904	2.919	2.881	2.955	2.952	**4.141**
Natural Questions	3.711	3.670	3.694	3.621	3.610	3.645	3.675	3.674	3.662	**3.756**	3.665	3.647	3.545
DuReader	4.044	3.991	3.978	3.973	3.932	3.891	4.051	3.846	3.742	3.967	3.776	3.780	**4.155**
TriviaQA (10M)	4.367	4.656	4.578	4.492	4.320	4.555	4.273	**4.740**	4.719	4.547	4.703	4.695	4.621
NarrativeQA	1.413	2.130	2.427	3.212	**3.427**	3.375	1.290	2.123	2.382	3.150	3.263	3.317	3.395
PopQA	2.810	3.347	3.396	3.268	3.380	3.376	2.787	3.298	3.305	3.337	3.384	3.362	**3.433**
2WikiMultiHopQA	2.646	3.579	3.582	1.855	3.381	3.583	1.339	3.263	3.445	1.651	3.332	3.541	**4.280**
HotpotQA	3.497	4.090	**4.225**	3.341	4.141	4.194	3.070	3.896	4.127	3.428	4.145	4.203	4.061
MuSiQue	1.988	2.462	**2.647**	1.801	2.522	2.605	1.704	2.317	2.258	1.895	2.462	2.614	2.211
Average	3.036	3.439	3.506	3.161	3.526	3.580	2.760	3.340	3.396	3.179	3.521	3.568	**3.760**

Table 8: Full F1 results: Comparison of MSA with same-backbone RAG baselines at all retrieval depths R@{1,5,10}. "RR" denotes RAG systems that perform both retrieval and reranking.

	Tokens	Qwen3-4B			Qwen3-4B (RR)			HippoRAG2			MSA (Ours)
Dataset		**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**@adaptive**
MS MARCO v1 (Bajaj et al., 2018)	7.34M	0.311	0.306	0.298	0.317	0.307	0.296	0.283	0.304	0.298	**0.687**
Natural Questions (Kwiatkowski et al., 2019)	1.47M	0.502	0.490	0.481	0.508	0.498	0.493	0.484	0.487	0.483	**0.534**
DuReader (He et al., 2018)	277K	0.517	0.482	0.490	0.531	0.484	0.492	0.314	0.278	0.289	**0.699**
TriviaQA (10M) (Joshi et al., 2017)	10M	0.668	0.718	0.703	0.719	0.728	0.748	0.695	0.732	0.749	**0.868**
NarrativeQA (Kočiský et al., 2018)	538K	0.213	0.330	0.372	0.509	0.451	0.448	0.209	0.249	0.262	**0.539**
PopQA (Mallen et al., 2023)	1.18M	0.245	0.266	0.267	**0.279**	0.271	0.260	0.256	0.272	0.270	0.228
2WikiMultiHopQA (Ho et al., 2020)	722K	0.146	0.542	0.558	0.168	0.540	0.561	0.141	0.569	0.594	**0.794**
HotpotQA (Yang et al., 2018)	1.35M	0.350	0.599	0.637	0.418	0.676	0.678	0.342	0.636	0.671	**0.717**
MuSiQue (Trivedi et al., 2022)	1.41M	0.114	0.257	0.285	0.143	0.290	0.290	0.128	0.272	0.305	**0.335**
Average		0.341	0.443	0.455	0.399	0.472	0.474	0.317	0.422	0.436	**0.600**

Table 9: Full F1 results: Comparison of MSA with SOTA RAG systems at all retrieval depths R@{1,5,10}. "RR" denotes RAG systems that perform both retrieval and reranking.

	Qwen3-235B			Qwen3-235B (RR)			Llama3.3			Llama3.3 (RR)			MSA (Ours)
Dataset	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**@adaptive**
MS MARCO v1	0.298	0.305	0.297	0.341	0.339	0.329	0.299	0.339	0.339	0.347	0.345	0.341	**0.687**
Natural Questions	0.558	0.541	0.542	0.555	0.548	0.550	0.591	0.585	0.580	**0.602**	0.579	0.567	0.534
DuReader	0.587	0.570	0.573	0.565	0.532	0.534	0.622	0.600	0.580	0.561	0.567	0.573	**0.699**
TriviaQA (10M)	0.737	0.771	0.769	0.740	0.737	0.767	0.763	0.818	0.842	0.799	0.839	0.839	**0.868**
NarrativeQA	0.195	0.272	0.297	0.450	0.451	0.435	0.176	0.281	0.309	0.463	0.449	0.444	**0.539**
PopQA	0.231	0.280	0.285	0.276	0.278	0.275	0.229	0.289	**0.293**	0.283	0.291	**0.293**	0.228
2WikiMultiHopQA	0.447	0.652	0.653	0.314	0.611	0.648	0.207	0.590	0.623	0.251	0.601	0.636	**0.794**
HotpotQA	0.567	0.708	**0.729**	0.555	0.714	0.726	0.499	0.667	0.713	0.559	0.709	0.721	0.717
MuSiQue	0.280	0.376	**0.412**	0.249	0.385	0.396	0.234	0.350	0.379	0.257	0.359	0.391	0.335
Average	0.433	0.497	0.506	0.449	0.511	0.518	0.402	0.502	0.518	0.458	0.527	0.534	**0.600**

I Training Signals and Gradient Flow

MSA is end-to-end *trainable*: retrieval and generation are optimized inside the same model and the same representation space. It is not, however, differentiable through the discrete selection step. Three training signals should be distinguished.

(1) Generation loss through the memory representations. The generation loss $\mathcal{L}_{\text{LLM}}$ is back-propagated through the attention over the selected compressed memory $\bar{K}_i, \bar{V}_i$ ($i \in \mathcal{I}$), through the chunk-wise mean pooling $\phi(\cdot)$, into the projections W_K^h, W_V^h and the document hidden states H_i. The memory representations consumed by attention are therefore trained by the answer loss, whereas in a retrieve-then-read pipeline the document encoder receives no gradient from it. Generative Retrieval, in which the model emits document IDs, is trained by the same token-level cross-entropy.

(2) No gradient through the Top-k decision. The index set $\mathcal{I} = \text{Top-}k(\{s_i\})$ is a hard selection and is not differentiated through. We use no straight-through estimator, no continuous relaxation, and no reinforcement-learning objective for this step.

(3) Router supervision. The Router Projectors $W^h_{Q^R}$ and $W^h_{K^R}$ are trained by the auxiliary loss $\mathcal{L}_{\text{aux}}$ (Eq. 4), which acts on the continuous relevance scores s_i and thus supervises which documents the Top-k step selects.

J MEMORY MANAGEMENT

MSA encodes every document independently (Independent Document Processing with document-wise RoPE), and the chunk-wise pooling never mixes tokens of two documents in one chunk. Memory entries therefore carry no cross-document dependency, which makes the memory bank editable at document granularity:

- **Insertion.** A new document is passed once through Stage 1 (Global Memory Encoding); its $(\bar{K}, \bar{V}, \bar{K}^R)$ chunks are appended to the bank without affecting any existing entry.
- **Deletion.** All chunks belonging to the document are removed. The removal is exact and requires no retraining, since the stored content does not reside in the model weights.
- **Replacement.** A deletion followed by an insertion.

This contrasts with parameter-based memory (e.g., LoRA or continual pre-training), where removing a single piece of knowledge becomes an unlearning problem, and with fixed-size recurrent states, whose contents cannot be addressed individually.

K ADDITIONAL ANALYSES

K.1 RETRIEVAL-SIDE METRICS

Table 10 reports evidence recall. Because MSA decides the number of retrieved documents adaptively through Generative Retrieval (Section 2.5), there is no fixed depth k for MSA; we report recall at its adaptive budget, which averages **1.34 documents per query**. With this budget MSA reaches an average recall of 0.754, above the same-backbone retriever at depth 5 (0.731) and below it at depth 10 (0.804), i.e., close to depth-10 recall while reading far fewer documents.

Localization and answer synthesis can also be separated with these numbers. On PopQA, MSA's recall (0.4845) is lower than the retriever's at depth 5 (0.4985), yet its answer score is higher (3.433 vs. 3.273, Table 2). We attribute this to the adaptive budget, which brings back fewer irrelevant documents than a fixed depth, and to the per-layer compressed KV, which provides partial information about relevant memory beyond the documents actually read.

Table 10: Evidence recall. MSA uses its adaptive retrieval budget (1.34 documents per query on average); the baseline is the Qwen3-4B-Embedding retriever at fixed depths.

Dataset	MSA @adaptive	Emb R@1	Emb R@5	Emb R@10
MS MARCO v1	0.8815	0.3377	0.8089	0.9399
Natural Questions	0.8654	0.9220	0.9867	0.9919
DuReader	0.9006	0.9094	0.9905	0.9959
TriviaQA (10M)	0.6237	0.2008	0.6855	0.8971
NarrativeQA	0.7713	0.1980	0.4232	0.5324
PopQA	0.4845	0.2412	0.4985	0.5120
2WikiMultiHopQA	0.8580	0.4338	0.7293	0.7638
HotpotQA	0.8500	0.4460	0.8645	0.9165
MuSiQue	0.5538	0.3196	0.5903	0.6869
Average	**0.754**	0.445	0.731	0.804

K.2 RAG WITH A RETRIEVER TRAINED ON OUR PRE-TRAINING CORPUS

To separate the architecture from exposure to our continual pre-training (CPT) data, we pre-train the same-backbone RAG retriever (Qwen3-4B-Embedding) on the same CPT corpus and re-run the RAG baseline (Table 11). The average LLM score at depth 10 moves from 3.242 to 3.256 (+0.014), whereas MSA reaches 3.760. The large drop in the ablation without CPT (Section 3.4) is thus better

explained by the router projectors, which are randomly initialized and first trained during CPT, than by a general advantage from the additional data.

Table 11: Same-backbone RAG with the original retriever and with the retriever pre-trained on our CPT corpus (LLM judge, 0–5).

	Qwen3-4B-Embedding			**CPT-trained Embedding**		
Dataset	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
MS MARCO v1	2.893	3.011	3.005	2.730	3.001	3.009
Natural Questions	3.452	3.374	3.297	3.401	3.401	3.387
DuReader	3.726	3.579	3.594	3.754	3.619	3.555
TriviaQA (10M)	4.133	4.414	4.273	4.148	4.391	4.266
NarrativeQA	1.611	2.567	2.860	1.945	2.963	3.225
PopQA	2.959	3.273	3.299	2.791	3.266	3.273
2WikiMultiHopQA	1.065	3.055	3.136	1.087	2.913	3.071
HotpotQA	2.252	3.582	3.787	2.251	3.424	3.770
MuSiQue	0.936	1.752	1.928	0.933	1.687	1.747
Average	2.559	3.179	3.242	2.560	3.185	3.256

K.3 Contamination Analysis

Queries and documents of the evaluation sets are filtered from our training data by exact match. As a stricter check we compute 13-gram overlap between the training data and the test data: query-side overlap is zero, and 1.26% of the test data shows 13-gram overlap on the document side.

K.4 Zero-Shot Evaluation on a Held-Out Benchmark

We evaluate MSA-4B zero-shot, without any fine-tuning, on LoCoMo (Maharana et al., 2024), a long-term conversational memory benchmark whose distribution appears in neither our continual pre-training nor post-training data (Table 12). RAG is evaluated only up to depth 3, because LoCoMo's corpus is short and deeper retrieval would amount to supplying nearly the full conversation. MSA outperforms RAG at comparable depth and scores below Mem0 (Chhikara et al., 2025), a purpose-built memory system; the Mem0 score is the overall LLM-as-a-judge result reported by Chhikara et al. (2025).

Table 12: Zero-shot results on LoCoMo (LLM-as-a-judge score).

Method	**Judge score**
RAG@2	54.8
RAG@3	61.5
Mem0 (Chhikara et al., 2025)	66.9
MSA-4B (zero-shot)	63.4

K.5 Routing Latency, Throughput, and Memory

Table 13 reports router latency, queries per second (QPS), and peak GPU memory at memory-bank sizes of 1M, 10M, and 100M tokens under three deployment configurations. The batch size in parentheses is per GPU. Router latency grows close to linearly with bank size (1.09, 8.9, and 85.5 ms on two GPUs), consistent with the $\mathcal{O}(L)$ routing cost in Appendix C; at 100M tokens the peak GPU memory stays within the 2×A800 node used in the paper, and sharding the routing keys over eight GPUs reduces both latency and per-device memory. At the smaller bank sizes QPS changes little because the GPUs are not fully occupied at these batch sizes; the batch size is held fixed across the three scales so that they remain directly comparable.

Table 13: Router latency, throughput, and peak GPU memory versus memory-bank size.

Memory bank	Configuration	Router latency	QPS	Peak GPU memory
1M	2 GPUs (batch 2)	1.09 ms	0.4476	10.96 GB
1M	8 GPUs (batch 2)	0.314 ms	1.057	11.56 GB
1M	8 GPUs (batch 16)	0.253 ms	8.058	12.37 GB
10M	2 GPUs (batch 2)	8.9 ms	0.4234	15.00 GB
10M	8 GPUs (batch 2)	2.81 ms	1.088	13.03 GB
10M	8 GPUs (batch 16)	2.53 ms	6.125	16.14 GB
100M	2 GPUs (batch 2)	85.5 ms	0.26	52.31 GB
100M	8 GPUs (batch 2)	22.31 ms	0.77	22.34 GB
100M	8 GPUs (batch 16)	22.91 ms	1.88	26.26 GB

K.6 Needle-in-a-Haystack beyond 1M Tokens

We extend the NIAH evaluation in Figure 4 from 1M to 100M tokens on the multi-query (MQ) and multi-key (MK1) subtasks of RULER (Hsieh et al., 2024) (Table 14).

Table 14: NIAH accuracy on RULER subtasks at memory-bank sizes beyond 1M tokens.

Memory bank	MQ	MK1
5M	98.68%	100%
10M	92.50%	100%
100M	71.00%	92.00%