CoMeT: Collaborative Memory Transformer for Efficient Long Context Modeling

Runsong Zhao[1,3*] **Shilei Liu**[3*] **Jiwei Tang**[2,3]
Langming Liu[3] **Haibin Chen**[3] **Weidong Zhang**[3] **Yujin Yuan**[3]
Tong Xiao[1†] **Jingbo Zhu**[1] **Wenbo Su**[3] **Bo Zheng**[3†]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]Tsinghua University [3]Future Living Lab of Alibaba
zhaors@mails.neu.edu.cn liushilei.lsl@taobao.com

Abstract

The quadratic complexity and indefinitely growing key-value (KV) cache of standard Transformers pose a major barrier to long-context processing. To overcome this, we introduce the **Co**llaborative **Me**mory **T**ransformer (CoMeT), a novel architecture that enables LLMs to handle arbitrarily long sequences with constant memory usage and linear time complexity. Designed as an efficient, plug-in module, CoMeT can be integrated into pre-trained models with only minimal fine-tuning. It operates on sequential data chunks, using a dual-memory system to manage context: a temporary memory on a FIFO queue for recent events, and a global memory with a gated update rule for long-range dependencies. These memories then act as a dynamic soft prompt for the next chunk. To enable efficient fine-tuning on extremely long contexts, we introduce a novel layer-level pipeline parallelism strategy. The effectiveness of our approach is remarkable: a model equipped with CoMeT and fine-tuned on 32k contexts can accurately retrieve a passkey from any position within a 1M token sequence. On the SCROLLS benchmark, CoMeT surpasses other efficient methods and achieves performance comparable to a full-attention baseline on summarization tasks. Its practical effectiveness is further validated on real-world agent and user behavior QA tasks. The code is available at: https://github.com/LivingFutureLab/Comet

1 Introduction

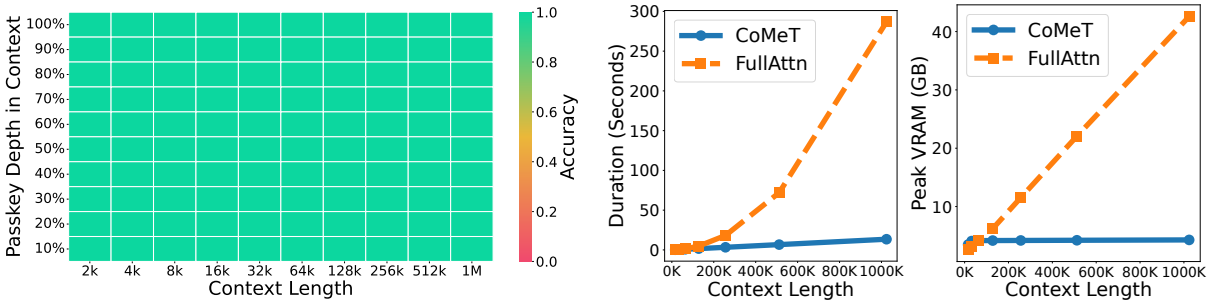
The ability to process and reason over vast contexts is a crucial frontier for Large Language Models (LLMs). From processing long documents for summarization (Huang et al., 2021; Pang et al., 2023) and question answering (Zhang et al., 2025a; Huang et al., 2021), to engaging in complex, multi-turn dialogues (Laban et al., 2025; Yi et al., 2024) and comprehending large codebases (Yuan et al., 2023), the capacity to capture long-range dependencies is a prerequisite for unlocking the full potential of LLMs in real-world applications. This requires models to not only understand but also persistently retain information across thousands or even millions of tokens, enabling them to grasp intricate narrative structures and make inferences based on a complete history.

However, the architectural foundation of modern LLMs, the Transformer (Vaswani et al., 2017), faces a fundamental scaling crisis when confronted with long sequences. Its standard implementation relies on a key-value (KV) cache that grows linearly with the input length, while the attention mechanism incurs quadratic computational complexity (as illustrated in Figures 1b and 1c). This makes processing extremely long contexts prohibitively expensive. To address this, two main categories of plug-and-play solutions have emerged. The first compresses the context into a shorter sequence (Mu et al., 2023b; Chevalier et al., 2023; Gao et al., 2024; Ge et al., 2024; Li et al., 2025, 2023; Tang et al., 2025a,b; Zhao et al., 2025; Liu et al., 2025). While effective, these methods are still constrained by the limits of information theory (Shannon, 1948): in general, the compressed length grow with the original. In typical settings, they mainly improve the constant factors in complexity rather than fundamentally changing its asymptotic behavior. The second category utilizes finite-state memory to achieve constant space and linear time (Dai et al., 2019; Rae et al., 2019; Bulatov et al., 2022; Rodkin et al., 2024; He et al., 2025). Yet, they struggle to retain fine-grained recent details, and often lack explicit gating mechanisms, making them prone to forgetting critical historical information.

To bridge this gap, we introduce the Collaborative Memory Transformer (CoMeT). As a parameter-efficient and non-invasive memory mod-

[*]Equal contribution.
[†]Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38384–38402



(a) Passkey Retrieval Accuracy (b) Inference time (c) GPU memory

Figure 1: CoMeT is trained on the passkey task (Munkhdalai et al., 2024) (i.e., the *Needle-in-a-Haystack* test) with a 32k context, yet it can retrieve a passkey from any position within a 1M-token context. Moreover, its inference time scales linearly with the context length, while GPU memory usage remains constant.

ule, CoMeT is specifically designed to overcome the limitations of prior finite-state models. Its core innovation lies in a synergistic memory system that explicitly addresses both the forgetting of critical information and the loss of recent details. To prevent forgetting, a fixed-size global memory employs a gated update mechanism to distill and shield salient historical information from being overwritten. Concurrently, a temporary memory managed by a First-In-First-Out (FIFO) queue captures fine-grained information from recent chunks, ensuring high-fidelity informational continuity. This design allows CoMeT to elegantly balance the retention of long-term memory with the awareness of long recent context. To enable efficient training on extremely long sequences, we introduce a layer-level pipeline parallelism strategy. This approach yields a 2.7× speedup over the naive context parallel method, making it feasible to fine-tune CoMeT on contexts up to 128k tokens using just 16×80GB GPUs.

The capabilities unlocked by CoMeT are substantial. Trained only on 32k-length sequences, CoMeT remarkably extrapolates to accurately retrieve a passkey from any position within a 1M token context (Figure 1a). This feat is achieved with a 21× inference speedup and a 10× smaller memory footprint compared to a full-attention baseline at that length. Beyond synthetic tasks, we conduct comprehensive evaluations of CoMeT on both academic language sequence processing tasks and real-world application scenarios. The experimental results demonstrate that CoMeT achieves the highest average score among the efficient methods we evaluated. Notably, on summarization tasks requiring comprehensive understanding, a CoMeT-enhanced model with a memory of just ∼2.5k tokens performs on par with a standard Transformer processing the full, uncompressed context. In summary, CoMeT presents an efficient, practical, and accessible solution to the long-context challenge, pushing the boundaries of what is possible for LLMs.

2 Related Work

The pursuit of efficient long-context modeling has evolved along three dominant paradigms: augmenting the standard Transformer with recurrence, developing novel recurrent architectures to replace attention, and compressing context into a more manageable size (Tay et al., 2022; Xiao and Zhu, 2023). CoMeT operates within the first paradigm, offers a practical alternative to the second, and fundamentally differs from the third in its complexity guarantees.

Recurrent Transformers. The chunk-level recurrence is initially introduced into Transformer by Transformer-XL (Dai et al., 2019), which caches hidden states from previous chunks to extend the model's receptive field. Building on this foundation, subsequent work has explored various enhancements. Some methods, like ERNIE-Doc (Ding et al., 2021), concatenate hidden states output at the same layer to grant the model a theoretical receptive field over all preceding content. Compressive Transformer (Rae et al., 2019) introduces a dual-queue mechanism to store a compressed representation of older states instead of discarding them. Others, such as RMT (Bulatov et al., 2022) and Memformer (Wu et al., 2022), use memory tokens to recurrently encode historical information chunk by chunk. More recent approaches have designed sophisticated memory structures, such as the associative memory in ARMT (Rod-

kin et al., 2024) and the hierarchical system in HMT (He et al., 2025). While these methods successfully achieve $\mathcal{O}(N)$ time and $\mathcal{O}(1)$ space complexity, they suffer from two key limitations that CoMeT addresses. First, many lack explicit gating mechanisms to protect important long-term memories from being overwritten by newer information. Second, they often treat all historical information uniformly, failing to preserve a high-fidelity, fine-grained record of recent events, which is crucial for tasks requiring immediate contextual awareness.

Recurrent Sequence Models. Another line of work is based on classic recurrent architectures, mainly including Linear Attention mechanisms and State Space Models (SSMs). Linear Attention (Katharopoulos et al., 2020) compresses historical key-value information into fixed-size states by removing exponential operations in attention and utilizing the associative property of matrix multiplication; S4 (Gu et al., 2022), S5 (Smith et al., 2023), LRU (Orvieto et al., 2023), RWKV4/5 (Peng et al., 2023), and RetNet (Sun et al., 2023) employ data-independent decay mechanisms, while recent advances such as HGRN1/2 (Qin et al., 2023, 2024), Mamba1/2 (Gu and Dao, 2024; Dao and Gu, 2024), RWKV6 (Peng et al., 2024), and GSA (Zhang et al., 2024) introduce data-dependent decay mechanisms. DeltaNet (Yang et al., 2024b) and Gated DeltaNet (Yang et al., 2024a) incorporate test-time training to enhance long-term memory capabilities. However, these recurrent sequence methods are specifically designed as architectural alternatives to Transformers and cannot be directly applied to existing pre-trained LLMs in a plug-and-play manner, requiring models to be trained from scratch and thus limiting their adoption in the current LLM ecosystem.

Context Compression. Compression methods aim to compress contexts into shorter sequences. Methods such as SelectiveContext (Li et al., 2023), LLMLingua (Jiang et al., 2023; Pan et al., 2024), LongLLMLingua (Jiang et al., 2024), and EXIT (Hwang et al., 2025) shorten contexts by removing unnecessary portions, while Nano-Capsulator (Chuang et al., 2024), CompAct (Yoon et al., 2024), and FAVICOMP (Jung et al., 2025) paraphrase contexts into more concise text. Beyond text-level compression, approaches such as GIST (Mu et al., 2023a), AutoCompressor (Chevalier et al., 2023), LLoCO (Tan et al., 2024), ICAE (Ge et al., 2024), 500xCompressor (Li et al., 2025), and Activation Beacon (Zhang et al., 2025b) compress contexts into shorter compressed embeddings or KV caches. However, under a fixed compression ratio, the length of the compressed sequence still grows linearly with the original context length. This fails to fundamentally alter the asymptotic order of spatiotemporal complexity and can only improve efficiency by reducing constant factors.

Notation	Meaning
τ	Index of the current input chunk.
i	Index of the current Transformer layer.
$\mathbf{H}_\tau^i$	Hidden states of the τ-th chunk at layer i.
$\mathbf{G}_\tau^i$	Global memory tokens for chunk τ at layer i.
$\mathbf{T}_\tau^i$	Temporary memory tokens for chunk τ at layer i.
$\mathbf{S}_\tau^i$	Persistent global state for chunk τ at layer i.
$\mathbf{C}_\tau^i$	Compression tokens for chunk τ at layer i.
$\mathbf{R}_\tau^i$	Readout tokens for chunk τ at layer i.
m	Number of readout tokens.
$\mathrm{TL}(\cdot)$	A single Transformer layer computation.
$\mathrm{RLA}(\cdot)$	Residual Low-Rank Adapter module.
d_{model}	Hidden dimension of the model.
r	Rank of the low-rank projection in the RLA.

Table 1: Notation used to describe the CoMeT architecture in Section 3 and Figure 2.

3 Method

In this section, we introduce the architecture and mechanisms of the Collaborative Memory Transformer (CoMeT). For clarity, Table 1 summarizes the key notations used to describe our model. We will first delineate the overall framework in Section 3.1, then provide a detailed exposition of the global and temporary memory mechanisms in Section 3.2, and finally, present our layer-level pipeline parallelism strategy for efficient distributed training in Section 3.3.

3.1 Overall Framework

Following prior work, CoMeT processes the input context in a chunk-by-chunk manner. As illustrated in Figure 2, at the i-th Transformer layer, when processing the τ-th input chunk, the model prepends the global memory $\mathbf{G}_\tau^i$ and temporary memory $\mathbf{T}_\tau^i$ to the chunk's hidden states $\mathbf{H}_\tau^i$. Through the causal self-attention mechanism, $\mathbf{H}_\tau^i$ can retrieve relevant information from both memories to inform next-token prediction. Concurrently, we interleave a set of compression tokens $\mathbf{C}_\tau^i$ within $\mathbf{H}_\tau^i$ to distill fine-grained local information. Finally, m readout tokens $\mathbf{R}_\tau^i$ are appended to the sequence to summarize the chunk's most salient content. The overall computation of

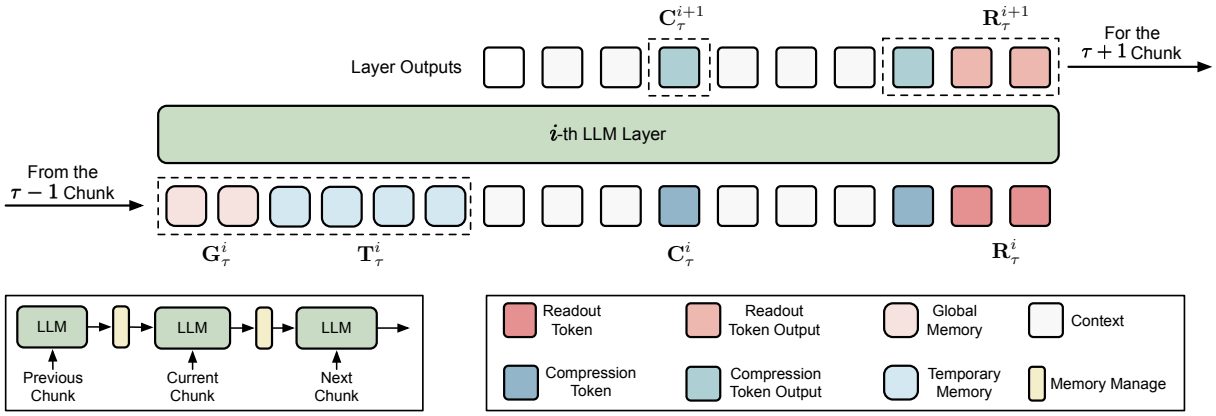


Figure 2: Overview of CoMeT. At layer i, the global memory $\mathbf{G}_\tau^i$ and temporary memory $\mathbf{T}_\tau^i$ are prepended to the current chunk's hidden states $\mathbf{H}_\tau^i$. Compression tokens $\mathbf{C}_\tau^i$ are interleaved within the hidden states for fine-grained information capture, while readout tokens $\mathbf{R}_\tau^i$ are appended at the end to distill key information for updating the global state. All tokens interact through causal self-attention, enabling the model to retrieve relevant historical information while processing the current chunk.

a single Transformer layer is thus formulated as: $\mathbf{H}_\tau^{i+1}, \mathbf{C}_\tau^{i+1}, \mathbf{R}_\tau^{i+1} = \mathrm{TL}(\mathbf{G}_\tau^i, \mathbf{T}_\tau^i, \mathbf{H}_\tau^i, \mathbf{C}_\tau^i, \mathbf{R}_\tau^i)$, where TL denotes the Transformer layer computation.

3.2 Collaborative Memory Mechanisms

CoMeT's memory system is composed of two synergistic components: a global memory for long-range dependencies and a temporary memory for recent context.

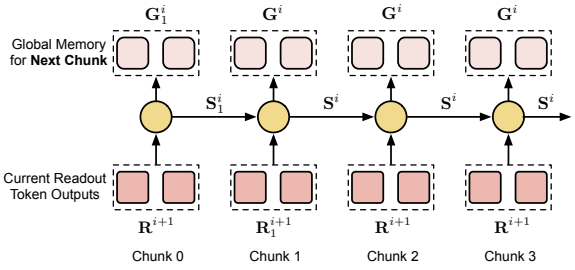


Figure 3: Architecture of the global memory mechanism. At each layer i and chunk τ, the global state $\mathbf{S}_\tau^i$ is transformed by a RLA to produce the global memory $\mathbf{G}_\tau^i$. The state is then updated for the next chunk via a gating mechanism that selectively integrates information from the normalized readout tokens $\mathbf{R}_\tau^{i+1}$.

Global Memory. As depicted in Figure 3, the global memory $\mathbf{G}_\tau^i$ is derived from a persistent global state $\mathbf{S}_\tau^i$. Our preliminary experiments reveal that introducing an excessive number of parameters for this state-to-memory transformation degrades performance. We therefore employ a parameter-efficient module we term the **Residual Low-Rank Adapter (RLA)**, which transforms a state vector by adding a low-rank projection:

$$\mathrm{RLA}(\mathbf{X}) = \mathbf{X} + \mathbf{W}_{\text{up}}(\mathbf{W}_{\text{down}}\mathbf{X}) \tag{1}$$

where the projection matrices are $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d_{\text{model}} \times r}$ and $\mathbf{W}_{\text{down}} \in \mathbb{R}^{r \times d_{\text{model}}}$. The global memory is thus computed as $\mathbf{G}_\tau^i = \mathrm{RLA}(\mathbf{S}_\tau^i)$. This additive, low-rank structure ensures minimal parameter overhead while promoting stable training. We set the rank $r = 8$ unless stated otherwise[1].

The global state for the next chunk $\mathbf{S}_{\tau+1}^i$ is updated using the output readout tokens $\mathbf{R}_\tau^{i+1}$. Prior to the update, these tokens are normalized to form a candidate state: $\tilde{\mathbf{S}}_{\tau+1}^i = \mathrm{RMSNorm}(\mathbf{R}_\tau^{i+1})$. We then employ a gating mechanism for the update:

$$\mathbf{S}_{\tau+1}^i = \mathbf{g} \odot \mathbf{S}_\tau^i + (\mathbf{1} - \mathbf{g}) \odot \tilde{\mathbf{S}}_{\tau+1}^i \tag{2}$$

where the gate $\mathbf{g} = \sigma(\mathbf{W}_{\text{g}}([\mathbf{S}_\tau^i; \tilde{\mathbf{S}}_{\tau+1}^i]))$. Here, $[\cdot;\cdot]$ denotes concatenation along the feature dimension, $\mathbf{W}_{\text{g}} \in \mathbb{R}^{2d_{\text{model}} \times 1}$ is a learnable weight matrix, and σ represents the sigmoid function. This mechanism allows the state to selectively absorb new information while shielding essential historical information from being overwritten. Furthermore, this additive update structure, reminiscent of gates in LSTMs and GRUs, creates a more direct path for gradient flow across chunks.

Temporary Memory. As shown in Figure 4, we manage the temporary memory $\mathbf{T}_\tau^i$ using a First-In-First-Out (FIFO) queue of fixed capacity. New

[1]The CoMeT module adds only 3.95M parameters (0.098%) to the Qwen3-4B-Instruct-2507 model when rank=8.

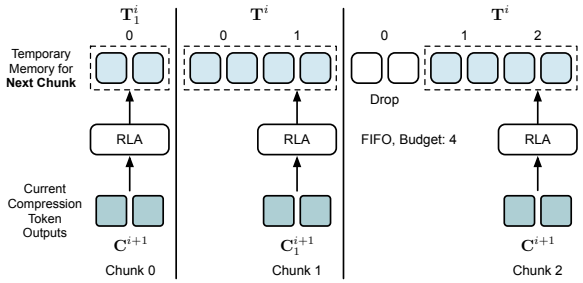


Figure 4: The architecture of the temporary memory mechanism. **CoMeT** employs a fixed-capacity FIFO queue to manage compressed representations of recent chunks. As new information from the current chunk is enqueued, the oldest memory entry is discarded. This rolling window of memory provides the model with a high-resolution view of the most recent context while maintaining a constant memory footprint.

memory entries are derived from the output compression tokens $\mathbf{C}_{\tau}^{i+1}$. These tokens are first processed by $\mathrm{RMSNorm}$ and then transformed using the same RLA module (as defined in Eq. 1) before being enqueued into the FIFO queue.

The FIFO nature of the queue preserves the temporal continuity of information from recent chunks. As a new entry is added, the oldest is discarded. This mechanism, combined with fine-grained compression, allows the model to maintain a high-resolution memory of the immediate context. From an optimization perspective, the FIFO queue also creates direct gradient paths back to recent chunks held in memory, enhancing training stability.

3.3 Efficient Long Context Training

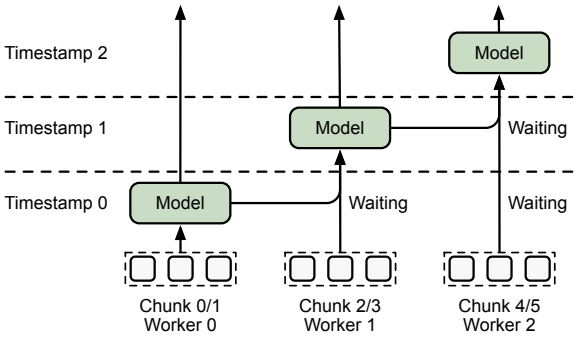


Figure 5: Naive context parallelism. Workers process chunks sequentially. Worker $j+1$ must wait for worker j to complete its entire computation before starting, creating a large pipeline bubble and leading to significant resource under-utilization.

Training CoMeT on extremely long sequences necessitates a distributed approach. A naive context parallelism strategy, as depicted in Figure 5, distributes chunks across GPU workers, with memory states passed between them via P2P communication. This method, however, suffers from a strict serial dependency, as each worker must wait for the previous one to complete its entire forward pass. This creates a large pipeline bubble, leaving most workers idle and leading to severe under-utilization of computational resources.

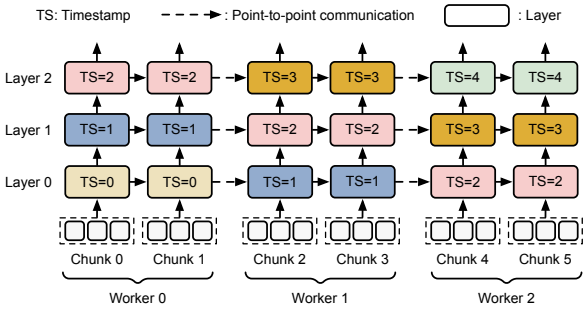


Figure 6: Our proposed layer-level pipeline parallelism. Computation and communication are interleaved at the layer level. Worker $j+1$ begins processing layer i as soon as it receives the necessary state from worker j, significantly reducing the pipeline bubble and maximizing hardware utilization.

To address this inefficiency, we propose a fine-grained pipeline parallelism method that interleaves computation and communication at the layer level (Figure 6). Rather than waiting for a full chunk computation, a worker, upon finishing layer i, immediately transmits the required memory state to the next worker. This enables the receiving worker to start on layer i while the sending worker concurrently advances to layer $i+1$. By maximizing worker concurrency, this strategy dramatically reduces idle time, boosts training throughput, and enables efficient scaling to very long sequences. This approach makes it feasible to train a Qwen3-4B-based **CoMeT** model with a 128K context length using just 16× 80GB GPUs. The visualization of pipeline bubbles for Layer-Level Pipeline Parallelism strategies is provided in Appendix A.

4 Experiments

To comprehensively evaluate CoMeT, we conduct experiments across three dimensions: (1) academic benchmarks to assess fundamental long-context language understanding capabilities, (2) real-world scenarios to validate practical applicability, and (3) passkey retrieval tasks to examine information extraction in extremely long contexts.

	Memory	GovRep	SumScr	QMSum	Qspr	Nrtv	QALT	CNLI	Avg
		R-1/2/L	R-1/2/L	R-1/2/L	F1	F1	F1	EM	
Full Attn	Full Context	52.7/17.1/20.3	19.1/4.2/10.2	16.3/4.6/10.1	3.5	2.5	3.8	0.0	7.87
Full Attn (FT)	Full Context	**61.0/31.9/33.0**	**32.5/7.6/19.0**	**37.4/12.9/25.6**	**40.3**	**22.1**	**64.2**	**89.1**	**42.23**
Compression									
LongLLMLingua	3072 tok	38.0/14.5/20.0	28.2/5.4/16.7	**34.6/11.4/23.3**	**35.7**	19.2	**65.9**	83.9	**37.36**
LLMLingua2	3072 tok	32.1/12.5/19.0	**29.8**/6.2/**17.9**	32.9/9.4/22.0	35.4	16.4	61.1	**88.2**	36.38
EXIT	3072 tok	48.6/21.3/24.2	28.8/5.8/17.4	32.3/8.9/21.4	35.4	14.9	59.9	86.5	36.94
ICAE	192×16 tok	25.4/5.5/17.4	21.2/3.3/13.9	28.7/7.8/20.6	18.5	15.7	54.9	74.2	29.04
500xCompressor	192×16 tok	34.4/12.4/20.9	23.5/4.4/14.9	24.1/7.1/18.1	23.0	19.0	56.3	82.6	32.54
ActivationBeacon	256×16 tok	**52.3/25.0/27.5**	28.0/**6.5**/17.1	31.8/10.2/22.7	33.5	**23.2**	56.8	25.8	30.71
Finite-state									
Transformer-XL	ws=5120	51.2/23.0/27.0	30.7/6.4/17.8	27.2/5.7/18.6	35.5	4.5	33.6	88.1	31.83
SWA	ws=5120	55.3/26.9/29.6	30.7/6.8/17.9	32.4/9.1/21.7	**39.1**	16.1	54.8	**88.3**	38.24
HMT	ms=3072	47.3/15.0/21.9	29.0/3.7/15.9	31.9/7.1/20.1	16.8	11.3	53.5	77.1	30.31
CoMeT	ms=2560	**62.5/31.1/33.4**	**33.4/8.3/19.8**	**35.6/12.0/24.6**	35.5	**22.6**	**56.0**	86.9	**40.10**
Avg. Length		10,535	8,617	13,291	5,462	19,250	6,085	2,210	

Table 2: Results on SCROLLS benchmark. All efficient methods use ∼3k memory budget. CoMeT outperforms other efficient methods and matches the fine-tuned full attention baseline on summarization tasks. GovRep, SumScr, QMSum, Qspr, Nrtv, QALT, and CNLI denote GovReport, SummScreenFD, QMSum, Qasper, NarrativeQA, QuALITY, and ContractNLI, respectively. FT denotes Fine-Tuned. SWA denotes Sliding Window Attention.

	2WikiMQA		HotpotQA	
	EM	**F1**	**EM**	**F1**
Full Attn	75.4	80.8	65.0	78.9
CoMeT	**75.5**	**81.0**	**65.9**	**80.0**
Avg. Length	1033		1443	

Table 3: Performance comparison on 2WikiMQA and HotpotQA. The last row shows the average context length for each dataset's development set.

4.1 Baseline Methods

We benchmark CoMeT against various plug-and-play methods, including context compression (e.g., LongLLMLingua, Activation Beacon) and finite-state models (e.g., Transformer-XL, Sliding Window Attention). Full Attention serves as the performance upper bound.

4.2 Experimental Setup

By default, our experiments employ Qwen3-4B-Instruct-2507 (Team, 2025a) as the backbone model. To ensure a fair comparison, all efficient methods are allocated a comparable memory budget of approximately 3k tokens. We fine-tune relevant models for 3 epochs on a 32k context length using a unified training configuration. Detailed parameters for each baseline and our training setup, including learning rates and optimizer settings, are provided in Appendix B.

4.3 Evaluation Results

Language Sequence Processing Tasks. We evaluate CoMeT on the SCROLLS benchmark (Shaham et al., 2022), which includes GovReport (Huang et al., 2021), SummScreenFD (Chen et al., 2022), QMSum (Zhong et al., 2021), Qasper (Dasigi et al., 2021), NarrativeQA (Kočiský et al., 2018), QuALITY (Pang et al., 2022), and ContractNLI (Koreeda and Manning, 2021). To assess performance on shorter sequences, we also include 2WikiMQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018). All fine-tunable models are trained for 3 epochs on a mixed dataset with up to 32k context length. Further details about these tasks are provided in Appendix C.

As shown in Table 2, CoMeT achieves the highest average score among all efficient methods. Crucially, on summarization tasks that require a holistic understanding of the input (GovRep, SumScr), CoMeT performs on par with the fine-tuned Full Attention baseline. Beyond the standard benchmark splits, we further evaluate CoMeT's extrapolation ability on long-context sequences exceeding its training length (Appendix D), where it continues to outperform SWA by a large margin. On shorter sequences (Table 3), CoMeT naturally matches Full Attention performance, as the entire input fits within a single chunk.

Real-World Application Scenarios. To demonstrate CoMeT's real-world utility, we evaluate it

	Memory	UQA	Terminal Bench
Full Attn	4k	51.3	–
Full Attn	32k	**81.3**	–
Full Attn	128k	–	**21.33**
xRAG	–	76.0	–
CoMeT	4k	**78.7**	–
CoMeT	5k	–	20.27

Table 4: Real-world application results on user behavior sequence QA and agent tasks. For user behavior QA, CoMeT outperforms the xRAG baseline and significantly improves over 4k truncated Full Attention. For code tasks, experiments are conducted at 128k sequence length with the Qwen3-8B model (8B instead of 4B, as the 4B model lacks the fundamental capabilities to solve the complex tasks in Terminal-Bench), where Full Attention training is enabled via Megatron-LM's (Shoeybi et al., 2019) sequence parallelism. CoMeT uses chunk size 4096 and memory size 1024 (G) + 4096 (T).

on two application-driven benchmarks: User Behavior QA (UQA) and a long-context agent task. Details are in Appendix C. The UQA benchmark requires reasoning over thousands of user interactions. On a real-world e-commerce dataset, CoMeT outperforms a strong industry xRAG[2] baseline by 2.7 accuracy points and a naive 4k Truncation baseline by 27.4 points (Table 4). For the agent task, we use iflow-cli[3] as the agent framework, fine-tune the model using 128k-token trajectories, and report results on Terminal-Bench (Team, 2025b). This extreme context length precludes training other efficient methods. Benefiting from our layer-level pipeline parallelism, CoMeT's training is 2.7× faster than naive context parallelism. It achieves performance competitive with a full-attention model while being vastly more efficient, validating CoMeT as a practical solution for deploying LLMs in real-world environments.

Passkey Retrieval Task. To evaluate CoMeT's performance in extreme-length contexts, we use a passkey retrieval task requiring finding a **5- to 7-digit** passkey within distractor text (details in Appendix E). After fine-tuning for 1500 steps on 32k-length sequences, CoMeT demonstrates remarkable extrapolation, successfully retrieving the passkey from any position within a 1M-token context (Figure 1a).

[2]xRAG encodes each behavior item in the user sequence into a single vector and concatenates these vectors with the question as input to the decoder.

[3]https://github.com/iflow-ai/iflow-cli

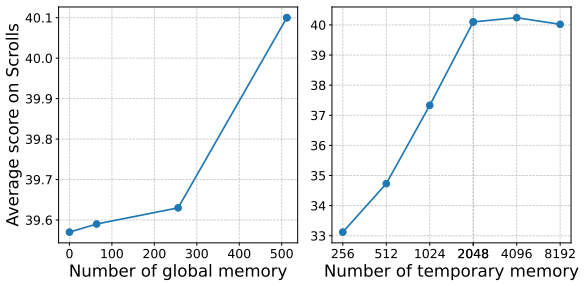
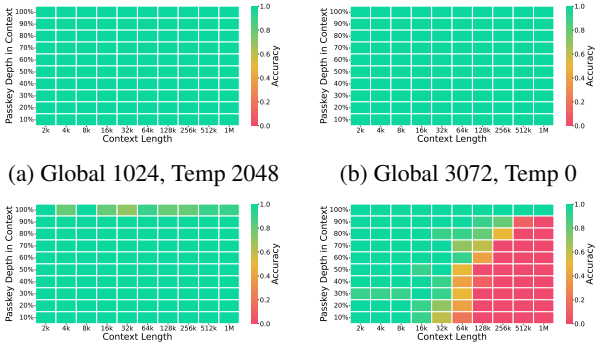


Figure 7: Performance impact of varying global and temporary memory sizes on the SCROLLS benchmark.



(a) Global 1024, Temp 2048 (b) Global 3072, Temp 0

(c) Global 8, Temp 3072 (d) Global 3072, no gate

Figure 8: Passkey retrieval accuracy under different memory configurations. (a) Balanced configuration with 1024 global and 2048 temporary memory. (b) Global-only configuration using all 3072 tokens for global memory. (c) Temporary-only configuration with minimal (8) global memory. (d) Global memory without gating mechanism, demonstrating the critical role of gates in long-term information retention.

5 Analysis

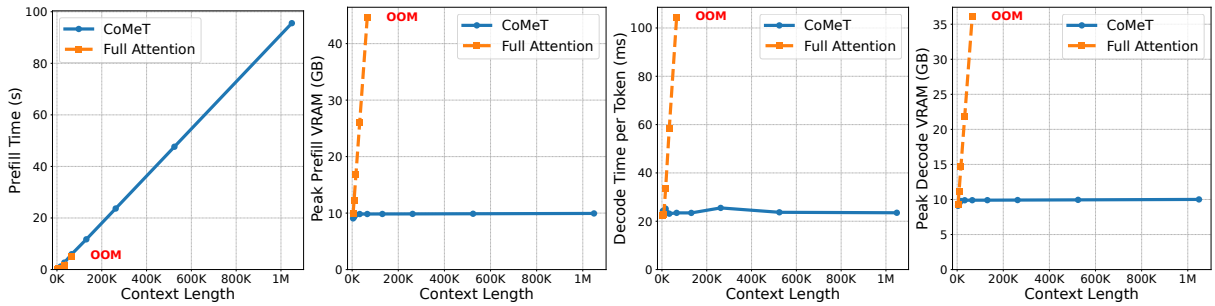
5.1 Generalization Across Model Families and Scales

To validate CoMeT's plug-and-play nature across diverse architectures and scales, we evaluate it on two additional models: Qwen3-14B and Llama-3.1-8B-Instruct. Results on the SCROLLS benchmark are presented in Table 5.

CoMeT consistently outperforms SWA across all three models, demonstrating broad generalizability. Notably, CoMeT's performance relative to full attention improves as model scale increases, retaining 95.0% of full attention performance on Qwen3-4B but 97.6% on Qwen3-14B. Furthermore, on the Llama-3.1-8B-Instruct model, CoMeT surpasses the full-attention baseline by 4.6%, suggesting that its benefits may be more pronounced in the Llama model family. These results confirm that CoMeT is an effective plug-and-play module for diverse model architectures and scales.

	Memory	GovRep	SumScr	QMSum	Qspr	Nrtv	QALT	CNLI	Avg
		R-1/2/L	R-1/2/L	R-1/2/L	F1	F1	F1	EM	
Qwen3-4B-Instruct-2507									
Full Attn (FT)	Full Context	61.0/**31.9**/33.0	32.5/7.6/19.0	**37.4/12.9/25.6**	**40.3**	22.1	**64.2**	**89.1**	**42.23** (100%)
SWA	ws=5120	55.3/26.9/29.6	30.7/6.8/17.9	32.4/9.1/21.7	39.1	16.1	54.8	88.3	38.24 (90.6%)
CoMeT	ms=2560	**62.5**/31.1/**33.4**	**33.4/8.3/19.8**	35.6/12.0/24.6	35.5	**22.6**	56.0	86.9	40.10 (95.0%)
Qwen3-14B									
Full Attn (FT)	Full Context	62.1/**32.9**/34.2	35.1/8.6/20.5	**37.7/13.3/25.5**	41.7	24.1	**72.3**	88.2	**44.18** (100%)
SWA	ws=5120	59.4/29.3/31.7	34.6/8.1/20.1	33.8/9.4/22.5	**41.9**	17.2	61.9	**88.6**	40.68 (92.1%)
CoMeT	ms=2560	**63.8**/32.5/**34.3**	**37.3/9.9/21.6**	36.7/12.3/24.7	37.7	**24.4**	67.5	**88.6**	43.14 (97.6%)
Llama-3.1-8B-Instruct									
Full Attn (FT)	Full Context	61.3/**32.1/33.4**	33.4/7.6/19.3	**36.6/12.3/24.6**	**37.0**	17.4	51.3	86.0	38.76 (100%)
SWA	ws=5120	60.2/31.0/32.2	33.3/8.1/19.6	34.0/10.7/22.6	36.3	15.9	44.0	86.3	37.04 (95.6%)
CoMeT	ms=2560	**62.9**/31.5/**33.4**	**36.6/9.5/21.1**	36.5/11.3/23.9	36.9	**23.6**	**54.7**	**87.3**	**40.55** (104.6%)

Table 5: Generalization results on SCROLLS benchmark across model families and scales. CoMeT consistently outperforms SWA across all settings. The percentage in parentheses denotes performance relative to the fine-tuned full attention baseline. CoMeT's relative performance improves with model scale (95.0% on Qwen3-4B vs. 97.6% on Qwen3-14B), and on Llama-3.1-8B-Instruct, CoMeT surpasses the full attention baseline by 4.6%.



(a) Prefill latency (b) Prefill memory usage (c) Per-token decode latency (d) Decode memory usage

Figure 9: Performance comparison of prefill and decode phases. (a) and (b) show the latency and memory usage during the prefill phase, while (c) and (d) present the per-token latency and memory consumption during the decode phase. Notably, the experiment is capped for Full Attention at 128k due to an OOM error, while our method (CoMeT) demonstrates scalability up to 1M.

5.2 Roles of Global and Temporary Memory

To dissect the distinct roles of our dual-memory system, we conduct ablation studies on memory allocation. We find that temporary memory is crucial for performance on in-domain sequence lengths, while global memory is paramount for extrapolation to out-of-domain lengths.

Temporary Memory Benefits Performance on In-Domain Lengths. On the SCROLLS benchmark, where tasks are within our 32k training length, temporary memory proves to be critical. As shown in Figure 7, overall performance improves with temporary memory size, saturating at 2,048 tokens. This demonstrates that temporary memory is vital for preserving the recent, detailed context. In contrast, increasing global memory offers only marginal gains here, suggesting its primary role lies elsewhere.

Global Memory Enables Length Extrapolation. The gated global memory is key for handling sequences longer than the training data. On the 1M-token passkey task (Figure 8), a global-only memory (Figure 8b) achieves perfect accuracy. In contrast, a configuration focused on temporary memory (Figure 8c) shows degraded performance, proving less effective at preserving a single fact over extreme distances. Most tellingly, removing the gating mechanism (Figure 8d) causes a complete performance collapse. This confirms the gate is essential for protecting key information.

5.3 Efficiency Analysis

We conduct an in-depth analysis of CoMeT's time and space efficiency during inference, comparing it with the Full Attention setting. Figure 9 presents the results based on the Qwen3-4B-Instruct-2507 model. CoMeT demonstrates superior space ef-

Rank	GovRep	SumScr	QMSum	Qspr	Nrtv	QALT	CNLI	Avg
	R-1/2/L	R-1/2/L	R-1/2/L	F1	F1	F1	EM	
4	62.4/31.3/**33.5**	33.4/7.8/19.4	35.2/11.1/24.0	33.9	21.7	54.0	86.1	39.18
8	62.5/31.1/33.4	**33.4/8.3/19.8**	**35.6/12.0/24.6**	**35.5**	**22.6**	56.0	86.9	**40.10**
64	**62.8/31.4/33.5**	32.5/7.7/19.1	**35.6**/11.7/24.4	34.2	21.9	**56.3**	**87.2**	39.79
512	62.6/**31.4**/33.4	31.2/7.0/18.7	35.3/11.2/24.2	34.2	20.5	54.2	86.7	39.02

Table 6: Ablation study on RLA rank on the SCROLLS benchmark. CoMeT (highlighted) uses $r = 8$ by default.

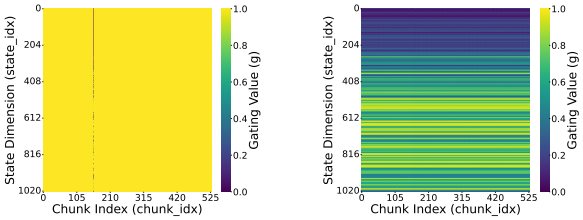
ficiency, maintaining constant peak memory consumption of $\sim$ 10GB regardless of context length (Figures 9b and 9d), while Full Attention's memory usage grows linearly, reaching OOM at 128k tokens. In terms of time efficiency, CoMeT's prefill latency scales linearly with context length (Figure 9a), and its per-token decoding latency remains stable at $\sim$ 22ms (Figure 9c). In contrast, Full Attention's decoding latency increases linearly, reaching 104ms at 65k tokens. Additional experiments with a smaller model further validate the theoretical complexity differences: CoMeT maintains linear prefill latency and constant memory consumption, while Full Attention shows quadratic growth in prefill latency and linear growth in peak memory (Figures 1b and 1c). These results demonstrate CoMeT's significant efficiency advantages in processing long contexts. For a more detailed analysis, please refer to Appendix F and Appendix G.

5.4 Sensitivity to Hyperparameters

Table 6 presents an ablation study on the RLA rank. Performance is largely stable across ranks from 4 to 512, with rank 8 achieving the best average score of 40.10. Increasing the rank to 512 leads to slight performance degradation. We therefore use $r = 8$ as the default setting. Other hyperparameters are determined based on our preliminary experiments, with detailed results provided in Appendix H.

5.5 Gating Value Visualization

To gain deeper insights into the role of the gating mechanism, we conduct a visualization analysis of CoMeT's behavior when processing extremely long texts. We select a 1M-token passkey retrieval task where the key is inserted at 30% depth. The analysis reveals that the gate is crucial for long-term retention, particularly in the model's deeper layers (e.g., 24, 28, 29, and 33). As illustrated in Figure 10a, upon encountering the passkey, the gate values in layer 33 drop to 0, allowing the critical information to be written into the global state. Subsequently, the gates close (values remain at 1),



(a) Layer 33 (b) Layer 7

Figure 10: Visualization of gating values when processing a 1M-token passkey retrieval task, where the passkey appears at chunk 157 (30% depth). The x-axis represents chunk indices and the y-axis represents the IDs of 1024 global memory states. (a) Layer 33 shows gate values dropping to 0 at chunk 157 when encountering the passkey, then consistently remaining at 1 to preserve the critical information. (b) Layer 7 exhibits differentiated gating patterns across states, indicating varied forgetting rates and multi-scale memory preservation.

effectively shielding this information from being overwritten by later chunks. In contrast, other layers exhibit more nuanced behavior. Figure 10b shows that different states within the same layer have varied gating patterns, suggesting they possess differentiated forgetting rates. This allows the model to preserve information across multiple time scales. Complete visualization results for all layers are provided in Appendix I.

6 Conclusion

In this work, we introduce CoMeT, a novel plug-in module that overcomes the scaling limitations of standard Transformers. By combining gated global memory for long-term dependencies with temporary FIFO memory for recent details, CoMeT achieves constant memory usage and linear time complexity. Remarkably, CoMeT trained on 32k contexts accurately retrieves information from 1M token sequences with 21$\times$ speedup over full attention. Combined with strong performance on the SCROLLS benchmark and proven real-world utility, CoMeT makes arbitrarily long-context processing practical for LLMs.

Limitations

While CoMeT effectively coordinates global and temporary memory, our current framework has not yet explored integration with episodic memory (test-time training) and external memory (such as notebooks and RAG-based knowledge bases). These components play crucial roles in human cognition for complex tasks. We view these not as fundamental flaws but as exciting avenues for future research. CoMeT's modular architecture provides a natural foundation for incorporating these additional memory types, and we hope our work will inspire further exploration in this direction.

Ethical Considerations and Societal Impact

CoMeT advances efficient long-context processing by significantly reducing memory and computational requirements, which can democratize access to large language models for researchers and practitioners with limited hardware resources. By enabling effective processing of very long sequences on modest infrastructure, CoMeT has the potential to facilitate personalized applications—such as user behavior understanding and intelligent assistants—that were previously accessible only to large organizations with substantial compute budgets.

However, the same capabilities that make CoMeT beneficial also warrant careful consideration of potential risks. The ability to efficiently process very long sequences of user interaction logs, as demonstrated in our UQA experiments, could be misused for large-scale surveillance or unauthorized analysis of personal behavior data. We encourage practitioners deploying CoMeT in such contexts to adhere to applicable data privacy regulations and to implement appropriate safeguards to protect user privacy.

Acknowledgments

This work was supported by Alibaba Group through Alibaba Research Intern Program, the National Natural Science Foundation of China (Nos. U24A20334 and 62276056), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.

Yu-Neng Chuang, Tianwei Xing, Chia-Yuan Chang, Zirui Liu, Xun Chen, and Xia Hu. 2024. Learning to compress prompt in natural language formats. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7756–7767, Mexico City, Mexico. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2978–2988.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10041–10071. PMLR.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Siyu Ding, Junyuan Shang, Shuohuan Wang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-doc: A retrospective long-document modeling transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2914–2927.

Jun Gao, Ziqiang Cao, and Wenjie Li. 2024. Selfcp: Compressing over-limit prompt via the frozen large language model itself. *Inf. Process. Manag.*, 61:103873.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*.

Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Zifan He, Yingqi Cao, Zongyue Qin, Neha Prakriya, Yizhou Sun, and Jason Cong. 2025. HMT: Hierarchical memory transformer for efficient long context language processing. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8068–8089, Albuquerque, New Mexico. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. 2025. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. *Preprint*, arXiv:2412.12559.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Dongwon Jung, Qin Liu, Tenghao Huang, Ben Zhou, and Muhao Chen. 2025. Familiarity-aware evidence compression for retrieval-augmented generation. *Preprint*, arXiv:2409.12468.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *Preprint*, arXiv:2505.06120.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Zongqian Li, Yixuan Su, and Nigel Collier. 2025. 500xCompressor: Generalized prompt compression for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25081–25091, Vienna, Austria. Association for Computational Linguistics.

Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Chunyang Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, and Jingbo Zhu. 2025. Autoencoding-free context compression for llms via contextual semantic anchors. *Preprint*, arXiv:2510.08907.

Jesse Mu, Xiang Li, and Noah Goodman. 2023a. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023b. Learning to compress prompts with gist tokens. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 101.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26670–26698. PMLR.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Ruhle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 963–981, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Bo Pang, Erik Nijkamp, Wojciech Kryscinski, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. 2023. Long document summarization with top-down and bottom-up inference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1267–1284, Dubrovnik, Croatia. Association for Computational Linguistics.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

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, and 13 others. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore. Association for Computational Linguistics.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, Przemysław Kazienko, Kranthi Kiran GV, Jan Kocoń, Bartłomiej Koptyra, Satyapriya Krishna, Ronald McClelland Jr., Jiaju Lin, Niklas Muennighoff, Fares Obeid, and 11 others. 2024. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. *Preprint*, arXiv:2404.05892.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. 2024. Hgrn2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*.

Zhen Qin, Songlin Yang, and Yiran Zhong. 2023. Hierarchically gated recurrent neural network for sequence modeling. In *Advances in Neural Information Processing Systems*, volume 36, pages 33202–33221. Curran Associates, Inc.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. 2024. Associative recurrent memory transformer. *Preprint*, arXiv:2407.04841.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. 2023. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Sijun Tan, Xiuyu Li, Shishir G Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E. Gonzalez, and Raluca Ada Popa. 2024. LLoCO: Learning long contexts offline. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17605–17621, Miami, Florida, USA. Association for Computational Linguistics.

Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. 2025a. Perception compressor: A training-free prompt compression framework in long context scenarios. In *NAACL (Findings)*, pages 4093–4108. Association for Computational Linguistics.

Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Jiaqi Chen, Lin Hai, Hai-Tao Zheng, and Hong-Gee Kim. 2025b. GMSA: enhancing context compression via group merging and layer semantic alignment. *CoRR*, abs/2505.12215.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey.

Qwen Team. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

The Terminal-Bench Team. 2025b. Terminal-bench: A benchmark for ai agents in terminal environments.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Qingyang Wu, Zhenzhong Lan, Kun Qian, Jing Gu, Alborz Geramifard, and Zhou Yu. 2022. Memformer: A memory-augmented transformer for sequence modeling. In *Findings of the association for computational linguistics: AACL-IJCNLP 2022*, pages 308–318.

Tong Xiao and Jingbo Zhu. 2023. Introduction to transformers: an nlp perspective.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2024a. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024b. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. 2024. Compact: Compressing retrieved documents actively for question answering. *Preprint*, arXiv:2407.09014.

Zhiqiang Yuan, Junwei Liu, Qiancheng Zi, Mingwei Liu, Xin Peng, and Yiling Lou. 2023. Evaluating instruction-tuned large language models on code comprehension and generation. *Preprint*, arXiv:2308.01240.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2025a. LongCite: Enabling LLMs to generate fine-grained citations in long-context QA. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5098–5122, Vienna, Austria. Association for Computational Linguistics.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2025b. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations*.

Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, Peng Zhou, and Guohong Fu. 2024. Gated slot attention for efficient linear-time sequence modeling. *ArXiv*, abs/2409.07146.

Runsong Zhao, Xin Liu, Xinyu Liu, Pengcheng Huang, Chunyang Xiao, Tong Xiao, and JingBo Zhu. 2025. Position IDs matter: An enhanced position layout for efficient context compression in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17715–17734, Suzhou, China. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

A Pipeline Bubble Visualization

To complement the discussion in Section 3.3, we provide a detailed visualization of pipeline bubbles for our proposed Layer-Level Pipeline Parallelism strategies. Figure 11 illustrates the execution timeline across four GPU ranks, where a sequence is divided into 4 segments[4] (one per GPU) and each GPU has an 8-layer model.

B Experimental Setup and Baseline Configurations

In our experimental setup, we employ Qwen3-4B-Instruct-2507 (Team, 2025a) as the backbone model by default. We uniformly set the memory budget to $\sim$ 3072 tokens for all baseline methods, except for the full attention which serves as the performance upper bound. Specifically, for text-level compression methods, we compress texts to 3,072 tokens, while texts shorter than 3,072 tokens remain uncompressed. For activation-level compression methods, given that the model is trained on sequences of 32k length, we set the chunk size to 2048 and employ 192 special tokens for compression per chunk. The configurations for other baseline methods are as follows: SWA adopts a window size of 5,120, Transformer-XL retains the most recent 3072 hidden states with a chunk size of 2,048, and HMT uses 32 sensory memory slots with a long-term memory budget of 3040. We configure CoMeT with 512 global memory, 2,048 temporary memory, a chunk size of 2,048, and insert one compression token every 8 context tokens, as this configuration achieves excellent performance without requiring the larger budget used by other baseline methods, as demonstrated in Table 7.

Unless otherwise noted, we adopt a unified training configuration for all methods requiring fine-tuning: batch size of 64 with sequences of varying lengths packed to 32k for training; learning rate of $5e-5$ with 10 warmup steps followed by cosine decay to 0; Adam optimizer with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For training-free methods, we directly evaluate their performance on the fine-tuned full attention model to assess the effectiveness of pure compression strategies.

C Dataset Construction Details

This appendix provides additional information on the construction of the mixed datasets used for fine-tuning our models, as mentioned in the main experiments section.

SCROLLS Mixed Dataset. To comprehensively evaluate the long-context processing capabilities of our models, we create a unified training and validation dataset derived from the SCROLLS benchmark (Shaham et al., 2022). This dataset amalgamates samples from all seven constituent tasks of SCROLLS: *GovReport*, *SummScreenFD*, *QMSum*, *Qasper*, *NarrativeQA*, *QuALITY*, and *ContractNLI*.

To manage training constraints and focus on a long-but-tractable context window, we filter the combined dataset to include only those examples where the total input sequence length does not exceed 32,768 tokens. The final dataset comprises 41,496 training samples and 7,455 validation samples. This process ensures that our training data is diverse, covering a wide range of tasks (summarization, question answering, natural language inference) and domains, while remaining within the specified maximum length for our fine-tuning process. During training, these variable-length sequences are packed into batches with a fixed total length of 32k tokens to maximize computational efficiency.

Shorter-Context QA Mixed Dataset. To ensure that our model's long-context adaptations do not degrade its performance on shorter sequences, we also construct a separate training set from established multi-hop Question Answering (QA) benchmarks. This dataset is created by sampling 20,000 examples from the *2WikiMQA* dataset and another 20,000 examples from the *HotpotQA* dataset. These 40,000 samples are then mixed to form a unified training set. Training on this mixed dataset allows the model to maintain its proficiency on tasks that require reasoning over shorter, more concise contexts, demonstrating that the CoMeT architecture does not compromise performance on standard-length inputs.

UQA Dataset. This dataset originates from a proprietary collection of user interaction logs from a major online e-commerce entity, which remains anonymous for confidentiality purposes. The core objective is to assess a model's ability to comprehend and reason over extended user activity sequences. The tasks designed for this dataset are diverse and include: (1) forecasting user interest in new product categories from their clickstream data; (2) providing tailored product recommendations

[4] If a segment is too long, it is further split into multiple chunks with chunk size within a GPU.

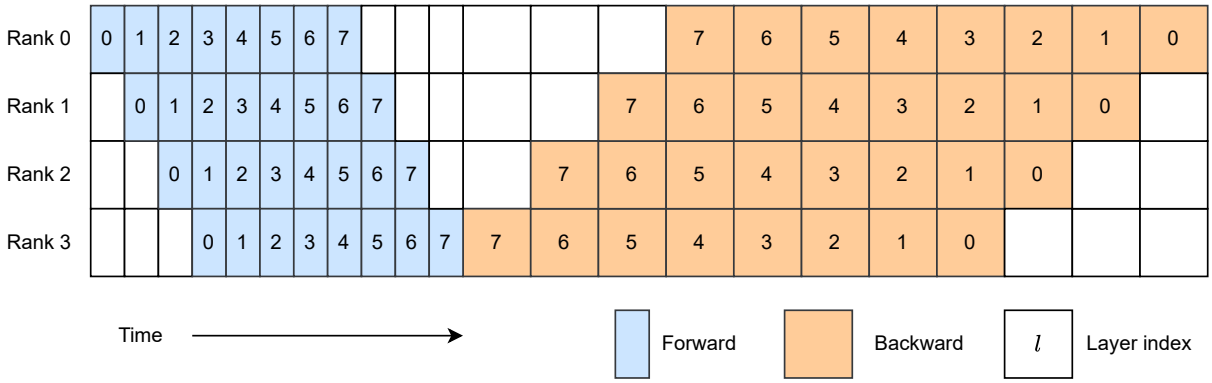


Figure 11: The visualization of pipeline bubbles for Layer-Level Pipeline Parallelism strategies. A sequence is divided into 4 segments (one per GPU) and each GPU has an 8-layer model. Blue cells indicate forward computation and orange cells indicate backward computation; white cells represent idle (bubble) time. The layer index l is shown within each cell. Compared to the naive strategy, our layer-level pipeline significantly reduces the bubble ratio.

informed by both click history and explicit search queries; and (3) synthesizing user behavior into a coherent summary. We employ an LLM-as-judge evaluation methodology, utilizing Qwen2.5-72B-Instruct as the judge model to assess the quality of model responses.

Long-Horizon Agent Dataset. This dataset is curated through a multi-stage, expert-driven methodology. Initially, we conduct a systematic analysis of GitHub issue forums to identify recurring and complex real-world software engineering problems. Subsequently, domain experts formulate a suite of tasks designed to emulate these challenges. Training trajectories are then generated by capturing the step-by-step interactions of an expert-operated, state-of-the-art agent model as it works to resolve these tasks.

D Extrapolation on Long-Context SCROLLS Subset

To evaluate CoMeT's extrapolation ability on realistic tasks, we tested our model (fine-tuned on 32k contexts) on a subset of the SCROLLS validation set containing only sequences **longer than 32k tokens**. This subset comprises 4,854 samples with an average length of 91k and a maximum length of 484k.

As shown in Table 7, CoMeT substantially outperforms Sliding Window Attention (SWA) on both tasks, with a particularly large gain on NrtvQA (13.2 vs. 5.8, a 128% improvement). This confirms that CoMeT's memory module continues to function effectively on sequences well beyond its training length. While CoMeT does not match Full Attention (FA) on this subset, it is worth noting that the base model (Qwen-3.4B-Instruct) natively supports a 256k context length, giving FA a significant advantage. Training CoMeT on longer contexts (e.g., 128k) via its layer-level pipeline parallelism may further close this gap.

Method	GovRep	NrtvQA
	R-1/2/L	F1
FA (Full Context)	**48.8 / 23.7 / 26.3**	**20.4**
SWA (ws=5120)	31.5 / 10.5 / 18.4	5.8
CoMeT (ms=2560)	38.6 / 14.1 / 19.9	13.2

Table 7: Performance on the long-context subset of SCROLLS (input $>$ 32k tokens; 4,854 samples, avg. 91k, max 484k). CoMeT is fine-tuned only on 32k contexts.

E Passkey Retrieval Task

Below we showed the chatML format of the passkey task.

> ["role": "system", "content": "You are an expert at finding a specific 'pass key' inside a long document. When asked, you will reply with only the pass key and nothing else.", "role": "user", "content": " There is an important info hidden inside a lot of irrelevant text. Find it and memorize them. I will quiz you about the important information there. To bake a cake, you need flour, sugar, and eggs. Mix them well. Bake at 350 degrees. (repeat x times) The pass key is **1392093**. Remember it. **1392093** is the pass key. To bake a cake, you need flour, sugar, and eggs. Mix them well. Bake at 350 de-

grees. (repeat y times) What is the pass key? The pass key is"]

F Detailed Efficiency Analysis

This section provides an in-depth analysis of the time and space efficiency of CoMeT during inference. We conduct experimental comparisons between the standard Full Attention architecture and the modified CoMeT architecture, focusing on system overhead during both the prefill and decode phases under varying context lengths. Figure 9 presents detailed comparative results based on the Qwen3-4B-Instruct-2507 model.

Space Efficiency. As illustrated in Figures 9b and 9d, CoMeT demonstrates remarkably superior efficiency. In both the prefill and decode phases, CoMeT maintains a constant peak memory consumption of approximately 10GB, remaining completely unaffected by context length growth. In contrast, Full Attention exhibits linear memory growth with increasing sequence length, encountering out-of-memory errors when processing 128k context tokens. This demonstrates that CoMeT's constant space complexity enables it to handle sequences of arbitrary length.

Time Efficiency. During the prefill phase (Figure 9a), CoMeT's latency scales linearly with context length, which aligns with its chunk-by-chunk processing mechanism. The advantage of CoMeT becomes even more pronounced in the decode phase. As shown in Figure 9c, the per-token decoding latency remains consistently stable at around 22ms regardless of context length. Conversely, Full Attention's decoding latency increases linearly with growing context, reaching 104ms at 65k tokens, nearly 5 times that of CoMeT, with this gap continuing to widen as sequence length increases.

To more clearly demonstrate the asymptotic complexity differences between the two architectures for longer sequences, we conduct supplementary experiments using a smaller model ($d_{model} = 768$, 12 layers). As shown in Figures 1b and 1c, the experimental results clearly validate our theoretical analysis: Full Attention exhibits quadratic growth in prefill latency and linear growth in peak memory, while CoMeT maintains linear prefill latency and constant memory consumption. Taken together, these experimental results convincingly demonstrate that CoMeT possesses overwhelming efficiency advantages in both time and space when processing long contexts, making it a robust solution for efficient long-sequence processing.

G Efficiency Comparison with RWKV

We benchmark CoMeT's latency against RWKV-7-Goose-World-2.9B-HF using the FLA library and the official Hugging Face implementation. Note that RWKV7 could not be tested beyond a context length of 32,768 due to a Triton/CUDA kernel error. Table 8 presents the results.

Model	Input Length	Prefill (ms)	Mem (MB)	decode (ms)
RWKV7	4096	136.8	7614	35.42
CoMeT	4096	235.3	9280	23.25
RWKV7	8192	282.6	9555	34.96
CoMeT	8192	519.3	9504	22.10
RWKV7	16384	375.2	13435	34.85
CoMeT	16384	1184.0	9954	24.20
RWKV7	32768	874.8	21195	34.58
CoMeT	32768	2678.9	10068	22.10
CoMeT	65536	5943.0	10074	22.50
...	...	...	...	...
CoMeT	1048576	95574.7	10176	22.50

Table 8: Latency comparison between CoMeT and RWKV7 at varying input lengths.

Two observations emerge from these results. First, both RWKV7 and CoMeT achieve **constant per-token decoding time** regardless of input length: CoMeT stabilizes at ∼22ms while RWKV7 stabilizes at ∼34ms. Second, CoMeT's **peak memory usage remains essentially constant** as context grows, whereas RWKV7's memory footprint increases with input length. We attribute RWKV7's faster prefill speed and its growing memory consumption to the parallel algorithm it employs during the prefill stage.

H Sensitivity to Hyperparameters

We conduct initial ablation studies on a different set of QA tasks to select our main hyperparameters before the SCROLLS experiments.

Chunk Size and Compression Token Interval. Table 9 reports results across varying chunk sizes, temporal strides, and temporal budgets. The key findings are:

1. A larger chunk size (*e.g.*, 2048) generally yields better performance, but with diminish-

ing returns and reduced efficiency. We chose 2048 as a balance.

2. A compression token interval of 8 (covering the last ∼16k tokens in temporary memory) performed better than an interval of 16 (covering ∼32k).

I Gating Value Visualization for All Layers

For completeness, we provide a comprehensive visualization of the gating values across all layers of CoMeT when processing the 1M-token passkey retrieval task (with the passkey inserted at 30% depth). Figure 12 presents the gating heatmaps for all 36 layers of the Qwen3-4B model.

J Analysis of Failure Cases and Information Loss

Following the qualitative analysis of failure cases on the **Qasper** dataset, where CoMeT showed a noticeable performance gap compared to the full attention baseline with the Qwen3-4B-Instruct-2507 model, we present representative failure patterns in Table 10.

Our key finding is that while CoMeT is susceptible to these types of information loss, **these failure modes are not unique to our method**. The Full Attention baseline exhibits similar errors, especially in long-range aggregation (as in case `00da...`). Furthermore, the performance gap on Qasper is model-dependent: when using Llama-3.1-8B-Instruct, the gap between CoMeT and Full Attention on Qasper narrows to a negligible **0.1 F1 points** (36.9 vs. 37.0).

Chunk Size	Global Memory	Temp Stride	Temp Budget	2WikiMQA EM/F1	HotpotQA EM/F1	NarrativeQA 32K EM/F1	NarrativeQA 64K EM/F1	NQ EM/F1
512	512	16	2048	64.4/71.1	55.9/71.3	12.8/32.3	12.4/27.8	68.3/78.0
512	512	8	2048	67.7/74.3	58.5/73.6	13.7/32.0	12.1/27.1	67.7/77.8
1024	512	16	2048	69.0/75.5	58.8/74.2	13.9/33.2	12.7/28.1	70.7/79.8
1024	512	8	2048	73.7/79.7	61.0/76.4	**15.5**/34.3	12.2/27.7	70.4/79.3
1024	512	8	4096	73.1/79.2	59.9/75.6	15.4/34.0	13.9/29.7	69.1/78.4
2048	512	8	2048	75.5/81.0	**65.9**/80.0	**15.5**/33.8	12.1/27.6	73.9/82.4
2048	512	8	4096	**76.6/82.1**	65.8/**80.1**	15.3/34.5	12.7/28.7	73.2/82.2
2048	512	16	4096	73.3/79.1	65.4/79.8	13.6/32.2	12.2/27.0	71.6/80.2
Full Attn	–	–	–	75.4/80.8	65.0/78.9	15.4/**35.5**	**15.9/32.0**	**77.3/84.2**

Table 9: Ablation on chunk size and compression token interval across QA tasks. Bold denotes the best result. This NarrativeQA is not from the SCROLLS dataset, hence the different results.

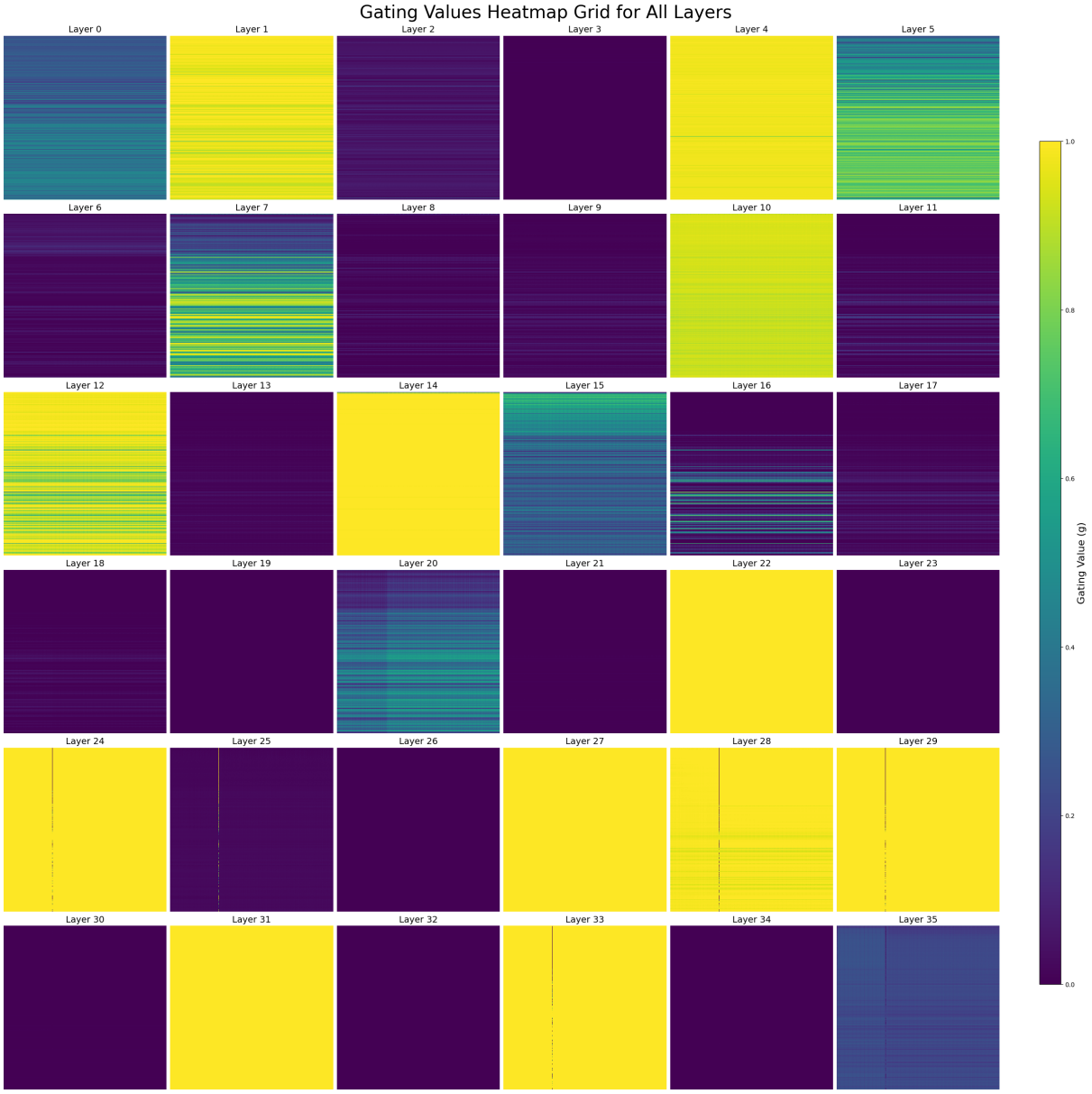


Figure 12: Complete visualization of gating values across all 36 layers when processing the 1M-token passkey retrieval task. Each subplot shows the gating heatmap for a specific layer, with the x-axis representing chunk indices and the y-axis representing the IDs of 1024 global memory states. The passkey appears at chunk 157 (30% depth).

Case ID	Question	CoMeT's Output	Full Attn's Output	Correct Answer	Failure Analysis
0416...	Is this hashtag prediction task an established task, or something new?	"Yes"	"Unanswerable"	"established task"	**Detail Loss & Oversimplification:** CoMeT correctly identified the affirmative nature of the answer but failed to extract the more informative core phrase, "established task." It provided the simplest possible correct response ("Yes"). This suggests its memory compression prioritized the binary judgment (yes/no) while discarding the specific descriptive details.
0407...	Which translation systems do they compare against?	"a state-of-the-art phrase-based MT system"	"hierarchical phrase-based system BIBREF29"	"English-Spanish MT system"	**Loss of Granularity & Generalization:** The original text uses the specific term "hierarchical phrase-based system." While Full Attention extracted this, CoMeT produced a generalized version. It retained the core concept ("phrase-based") but lost the more precise modifier ("hierarchical"). This indicates CoMeT's memory mechanism may be less sensitive to specific, descriptive terms, opting for more generic concepts and resulting in a loss of information granularity.
02f1a...	Why are current ELS's not sufficiently effective?	"due to low precision rates BIBREF11 and design challenges in training datasets BIBREF12"	"the extracted linked entities may be too ambiguous and coarse to be considered relevant to the summary"	"Linked entities may be ambiguous or too common"	**Causal Reasoning Failure:** The text establishes a causal chain: the problem is "ambiguity and coarseness," which is caused by "low precision rates..." The question asks for the problem itself. Full Attention correctly identified the problem. CoMeT, however, skipped the problem and extracted its cause. This suggests CoMeT's compressed context flattened the causal relationship, making it unable to distinguish cause from effect when answering a "Why" question.
00da...	What dataset is used?	"WMT 2014 shared translation task"	"Europarl corpus"	"Europarl corpus, WMT newstest 2014, News-Commentary-11, Wikipedia from WMT 2014, Multi-UN, EU-Bookshop, Rapid, Common-Crawl (WMT 2017)"	**Information Aggregation Failure:** The correct answer required aggregating a list of datasets from multiple, non-contiguous parts of the document. Both models failed to retrieve the complete list. This failure mode is particularly challenging for an efficient model like CoMeT. Its limited or compressed view of the global context makes it more prone to finding only the first or most locally salient answer piece, thus failing to perform the necessary long-range search and aggregation required to construct a comprehensive answer.

Table 10: Representative failure cases of CoMeT on the Qasper dataset.