ICLR 2025

MELODI: Exploring Memory Compression for Long Contexts

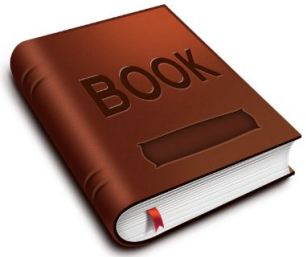
Yinpeng Chen, DeLesley Hutchins, Aren Jansen,

Andrey Zhmoginov, David Racz, Jesper Andersen

Google DeepMind

Problem:

Efficiently process long documents using short context windows.

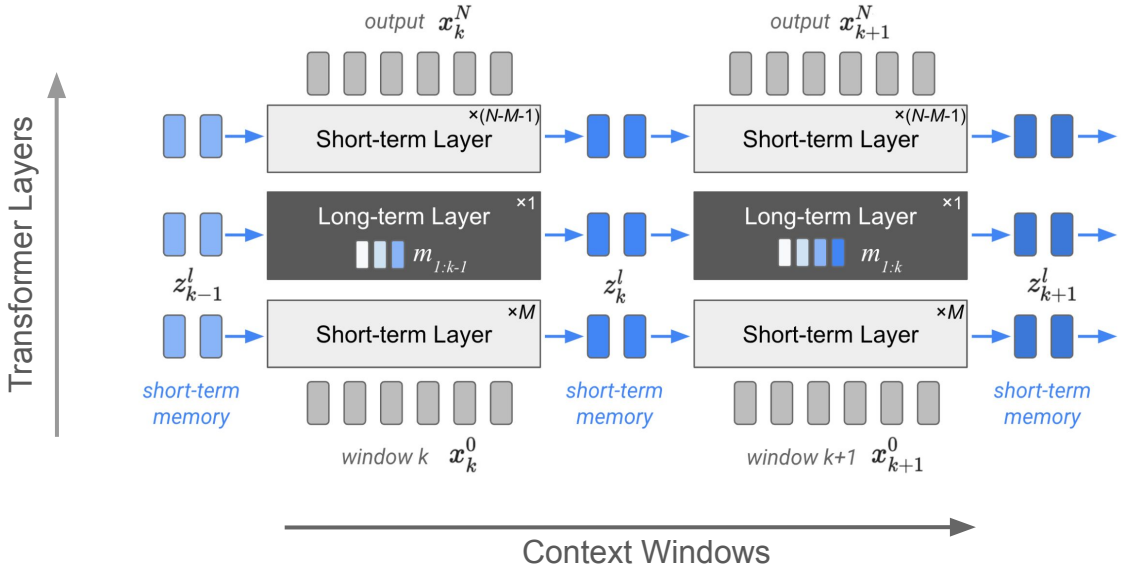


1 Long Window

Multiple Short Windows

Key Idea:

*Represent **short-term** and **long-term memory** as a **hierarchical compression** scheme that spans both transformer layers and context windows.*

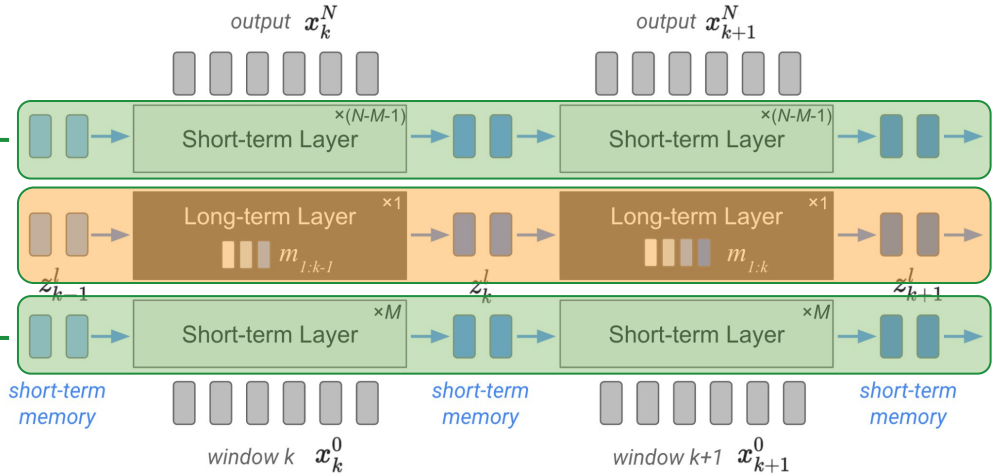


Sandwich Architecture

Short-term:

recurrent compression

across multiple layers



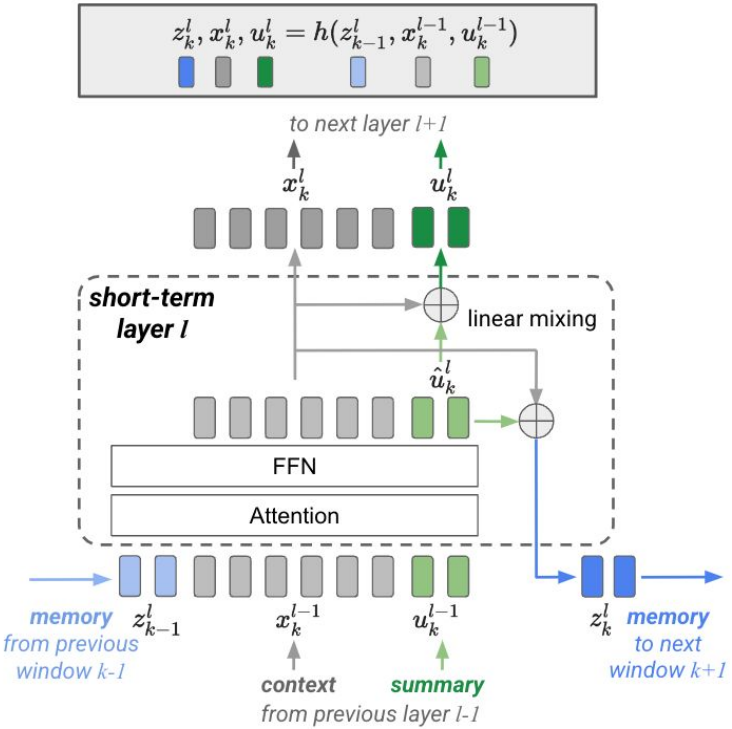
Long-term:

long-range compression

a single middle layer

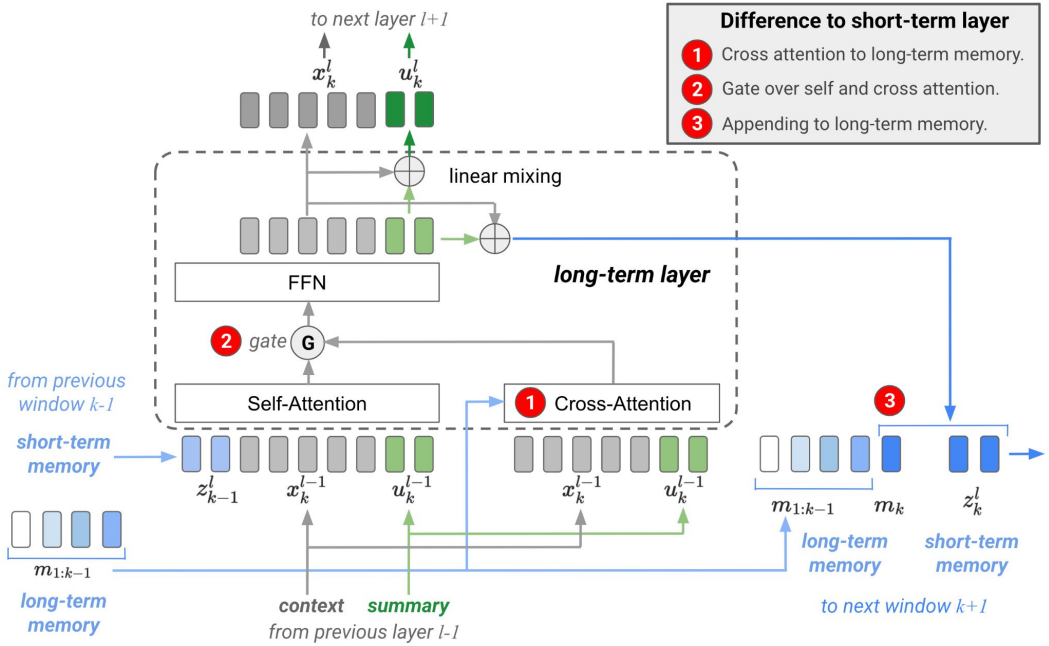
Short-Term Memory Layer

- The (long) input sequence is divided into (shorter) **blocks** or **context windows** of tokens.
- Each block is compressed into **summary tokens,** which are passed over **multiple layers**.
- Compression is **recurrent** across sequence length. Each block can attend to the summary tokens of the previous block.
- A **token mixing** operation creates separate **branches** of the summary tokens for the next layer, and for the next block.



Long-Term Memory Layer

- There is a **single** long-term layer in the middle of network stack.
- The long-term layer stores a KV-cache of the summary tokens from **multiple** previous blocks in a **FIFA queue**.
- The long-term layer uses a **higher compression rate** than the short-term layers, i.e. fewer summary tokens per block.
- It does **self-attention** over the current block of tokens, and **cross-attention** to the long-term KV-cache. A **gate** combines self and cross-attention.



Long-Term vs. Short-Term Memory

PROPERTY	LONG	SHORT
Number of layers	*single*	*multiple*
Update per window	*incremental*	*recurrent*
Capacity	$L \times Q_{max}$	$S \times N$

L: Number of long-term tokens per window.
S: Number of short-term tokens per window.
Q_{max}: Number of windows in the long-term queue.
N: Number of layers.

Experiments

Datasets: PG-19, arXiv Math, C4 (4K+).

Task: standard autoregressive language modeling.

Setup:

- Decoder-only Transformer (12 layers).
- Embedding dim 1024, 8 attention head (dim 128).
- FFN hidden layer dim 4096.
- Sequence length (per batch) 4096.
- 8 context windows (512 tokens per window).

Main Results (Perplexity ↓)

MODEL	MEMORY			PG19			arXiv		C4(4K+)
	All	Short	Long	Meena	T5	Custom	Meena	Custom	Custom
12 LAYERS									
Transformer XL	12.6M	12.6M	0M	8.76	11.54	12.63	2.61	3.23	18.61
Block Recurrent	12.1M	12.1M	0M	8.47	11.12	12.11	2.27	2.73	18.27
MELODI S_{192}+L_{32}	**10.8M**	2.4M	8.4M	**8.22**	**10.66**	**11.66**	**2.13**	**2.55**	**18.03**
Memorizing Trans.	146.8M	12.6M	134.2M	8.15	10.74	11.68	2.15	2.57	17.88
MELODI S_{128}+L_{64}	**18.4M**	1.6M	16.8M	8.16	10.61	11.66	2.13	2.55	18.01
MELODI S_{192}+L_{96}	27.6M	2.4M	25.2M	**8.08**	**10.48**	**11.47**	**2.11**	**2.51**	**17.75**

Clearly outperforming baselines.

Scaling up Well

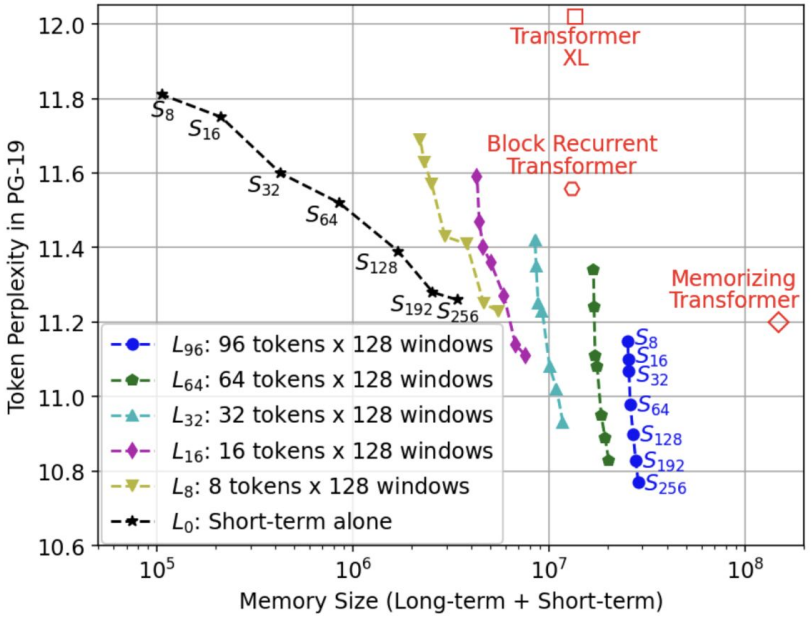
Scaling up model size

MODEL	#Layers	Embedding Dim	Model Size	Memory Size	Perplexity ↓
Transformer XL	12	1024	151M	12.6M	11.54
Memorizing Transformer	12	1024	151M	146.8M	10.74
MELODI $S_{128} + L_{64}$	12	1024	153M	18.4M	10.61
MELODI $S_{192} + L_{96}$	12	1024	153M	27.6M	**10.48**
Transformer XL	24	1024	302M	25.2M	10.15
Memorizing Transformer	24	1024	302M	159.4M	9.45
MELODI $S_{128} + L_{64}$	24	1024	306M	20.0M	9.30
MELODI $S_{192} + L_{96}$	24	1024	309M	30.0M	**9.16**
Transformer XL	36	1024	453M	37.8M	9.61
Memorizing Transformer	36	1024	453M	172.0M	8.92
MELODI $S_{128} + L_{64}$	36	1024	459M	21.6M	8.81
MELODI $S_{192} + L_{96}$	36	1024	463M	32.4M	**8.70**
Transformer XL	16	1536	453M	25.2M	9.69
Memorizing Transformer	16	1536	453M	226.5M	9.01
MELODI $S_{128} + L_{64}$	16	1536	456M	28.3M	8.89
MELODI $S_{192} + L_{96}$	16	1536	459M	42.5M	**8.79**

Scaling up sequence and window length

MODEL	Sequence Length	Window Length	Memory Size	Perplexity ↓
Transformer XL	4096	512	12.6M	11.54
Memorizing Transformer	4096	512	146.8M	10.74
MELODI $S_{128} + L_{64}$	4096	512	18.4M	10.61
MELODI $S_{192} + L_{96}$	4096	512	27.6M	**10.48**
Transformer XL	4096	1024	25.2M	11.26
Memorizing Transformer	4096	1024	159.4M	10.64
MELODI $S_{256} + L_{128}$	4096	1024	20.0M	10.47
MELODI $S_{384} + L_{192}$	4096	1024	30.0M	**10.36**
Transformer XL	8192	1024	25.2M	11.18
Memorizing Transformer	8192	1024	159.4M	10.42
MELODI $S_{256} + L_{128}$	8192	1024	20.0M	10.27
MELODI $S_{384} + L_{192}$	8192	1024	30.0M	**10.19**
Transformer XL	8192	2048	50.3M	10.94
Memorizing Transformer	8192	2048	184.5M	10.38
MELODI $S_{512} + L_{256}$	8192	2048	23.1M	10.20
MELODI $S_{768} + L_{384}$	8192	2048	34.6M	**10.10**

Ablation 1: Short-term and Long-term memories are **complementary**.



	L_0 0.0M	L_8 2.1M	L_{16} 4.2M	L_{32} 8.4M	L_{64} 16.8M	L_{96} 25.2M
S_8 0.1M	11.81	11.69	11.59	11.42	11.34	11.15
S_{16} 0.2M	11.75	11.63	11.47	11.35	11.24	11.10
S_{32} 0.4M	11.60	11.57	11.40	11.25	11.11	11.07
S_{64} 0.9M	11.52	11.43	11.36	11.23	11.08	10.98
S_{128} 1.7M	11.39	11.41	11.27	11.08	10.95	10.90
S_{192} 2.6M	11.28	11.25	11.14	11.02	10.89	10.83
S_{256} 3.4M	11.26	11.23	11.11	10.93	10.83	10.77

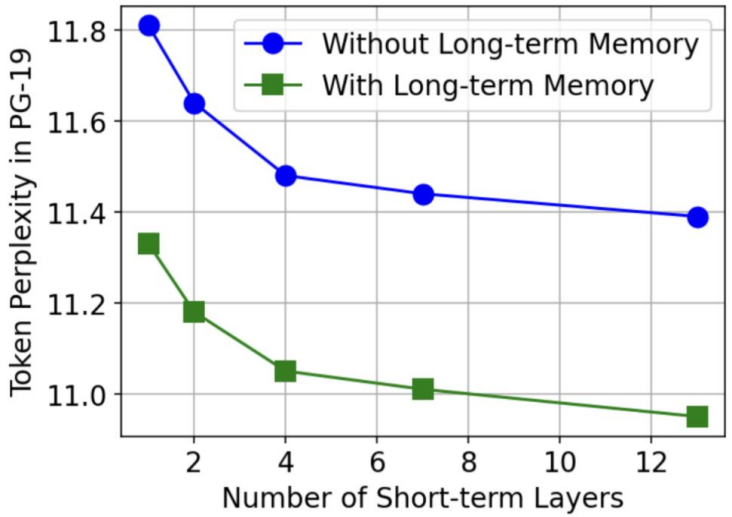
Ablation 2: Summary branching (**token mixing**) is helpful.

`ST`: short-term memory alone.
`ST+LT`: both short-term and long-term memory.

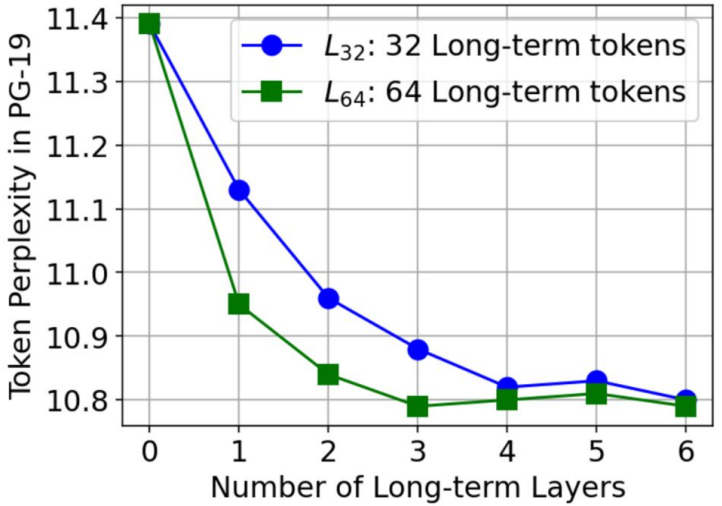
Branching	ST	ST+LT
No	11.68	11.24
Yes	**11.39**	**10.95**

Ablation 3: **Multiple** short-term layers + **One** long-term layer.

Multiple short-term layers are necessary.



One long-term layer is good enough.



Conclusion:

- **Hierarchical compression scheme (short-term + long-term memory).**
  - Short-term: combines sliding window attention with recurrent compression.
  - Long-term: stores compressed information from the entire sequence history.
- **Short-term and long-term memory are complementary.**
- **Very long effective context length, but requires far less memory.**