Long-Context Language Models

Sam Pan

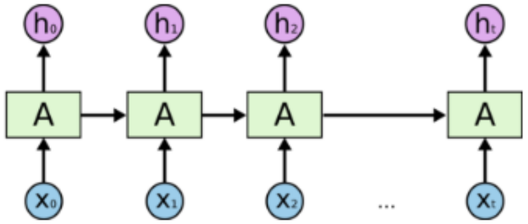
LongNet: Scaling Transformers to 1B Tokens

Jiayu Ding[♣*] **Shuming Ma**[♠*] **Li Dong**[♠] **Xingxing Zhang**[♠]
Shaohan Huang[♠] **Wenhui Wang**[♠] **Nanning Zheng**[♣†] **Furu Wei**[♠†]
[♠]Microsoft Research
[♣]Xi'an Jiaotong University

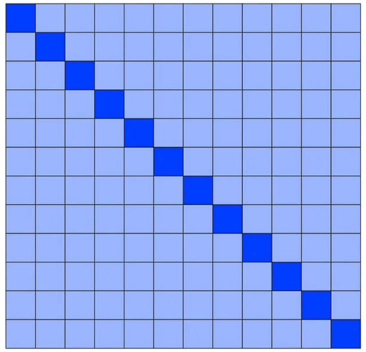
2023

Self Attention

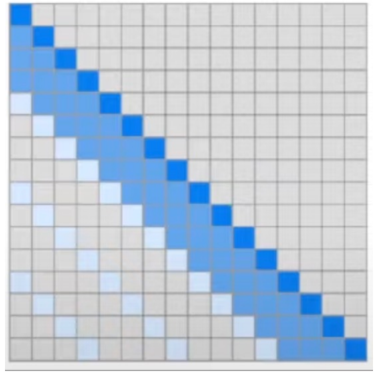
Recurrent



Vanilla Attention



Sparse Attention

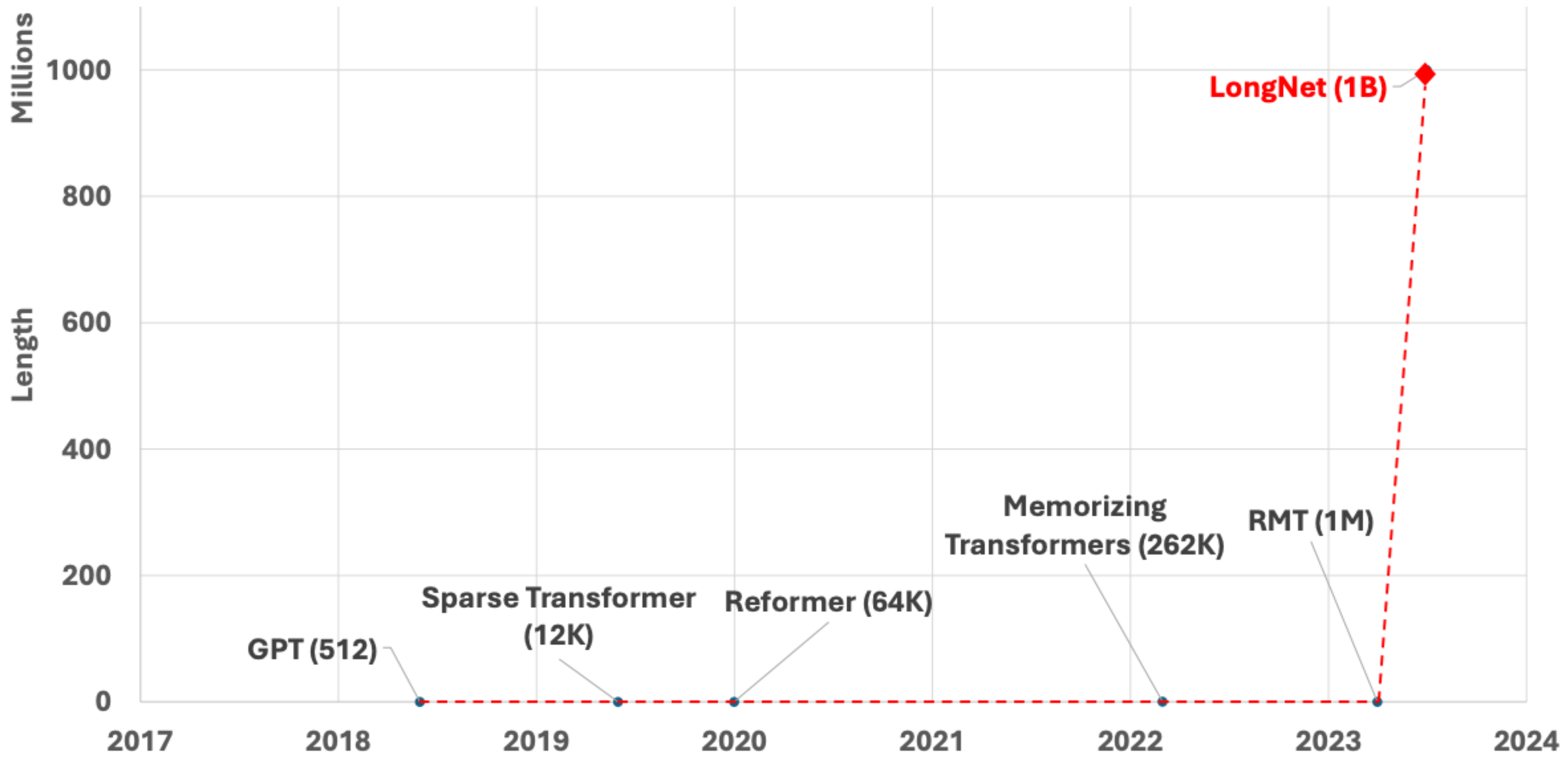


Self Attention

Linear Complexity essential for mass scaling

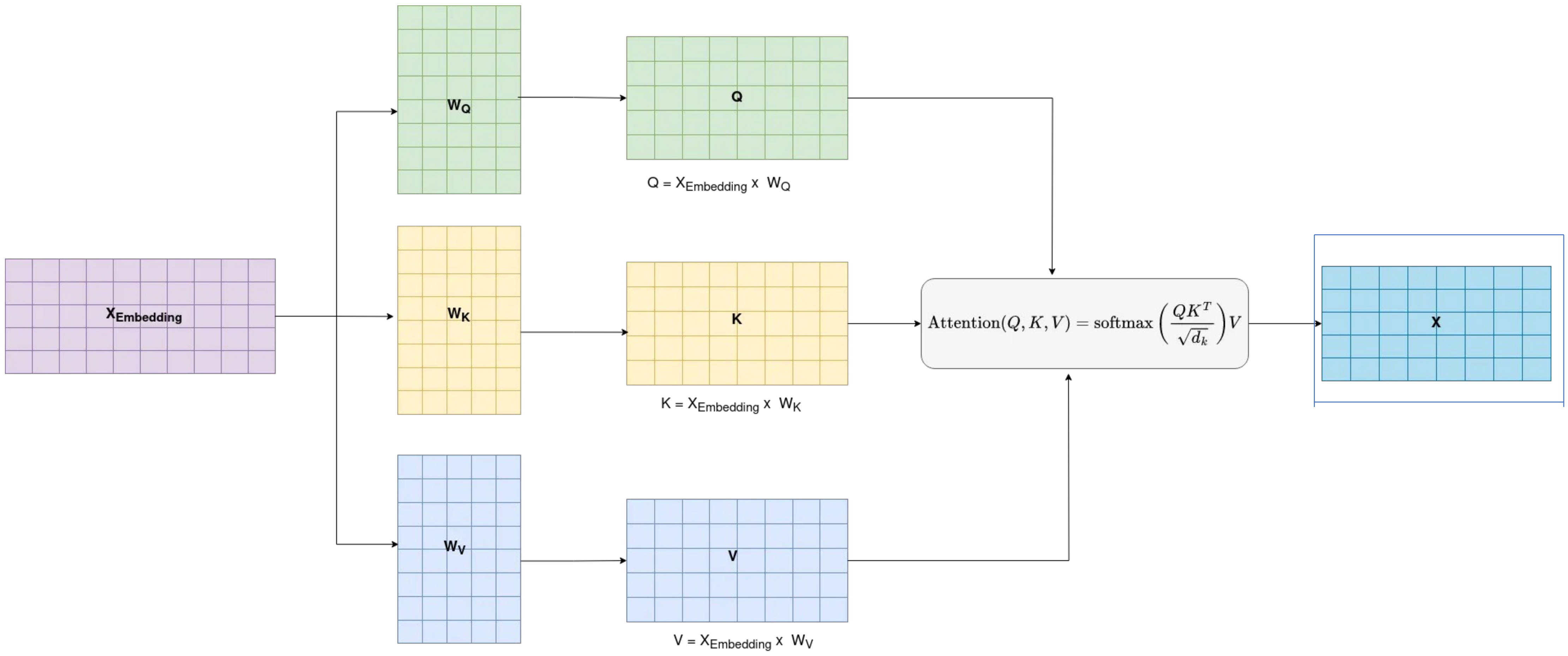
Method	**Computation Complexity**
Recurrent	$\mathcal{O}(Nd^2)$
Vanilla Attention	$\mathcal{O}(N^2d)$
Sparse Attention	$\mathcal{O}(N\sqrt{N}d)$
Dilated Attention (This Work)	$\mathcal{O}(Nd)$

Trend of Transformer sequence lengths

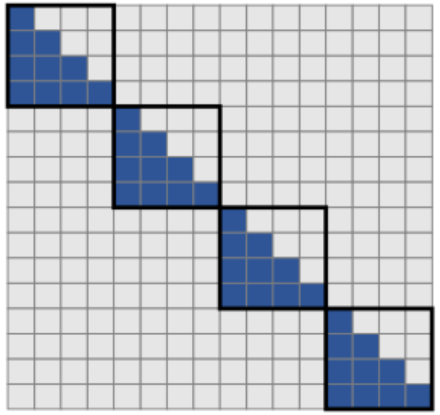


Sequence length, as the last atomic dimension of the neural network, is desirable to be unlimited

Vanilla Attention

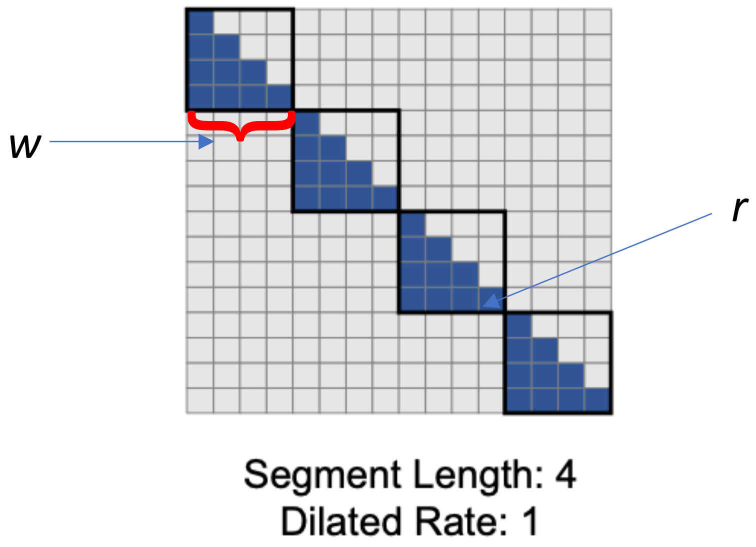


LongNet: Dilated Attention



Segment Length: 4
Dilated Rate: 1

LongNet: Dilated Attention

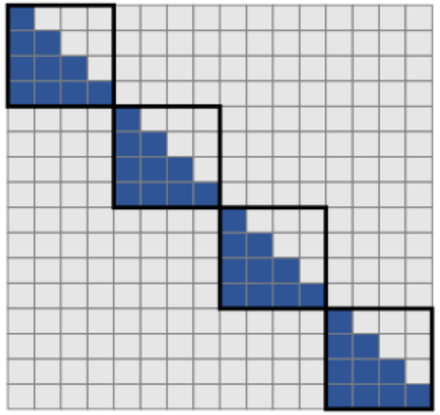


Introduces new parameters *w* and *r*

w – segment length

r – dilation rate

LongNet: Dilated Attention



w Segment Length: 4
r Dilated Rate: 1

$$\widetilde{Q}_i = [Q_{iw}, Q_{iw+r}, Q_{iw+2r}, \ldots, Q_{(i+1)w-1}]$$

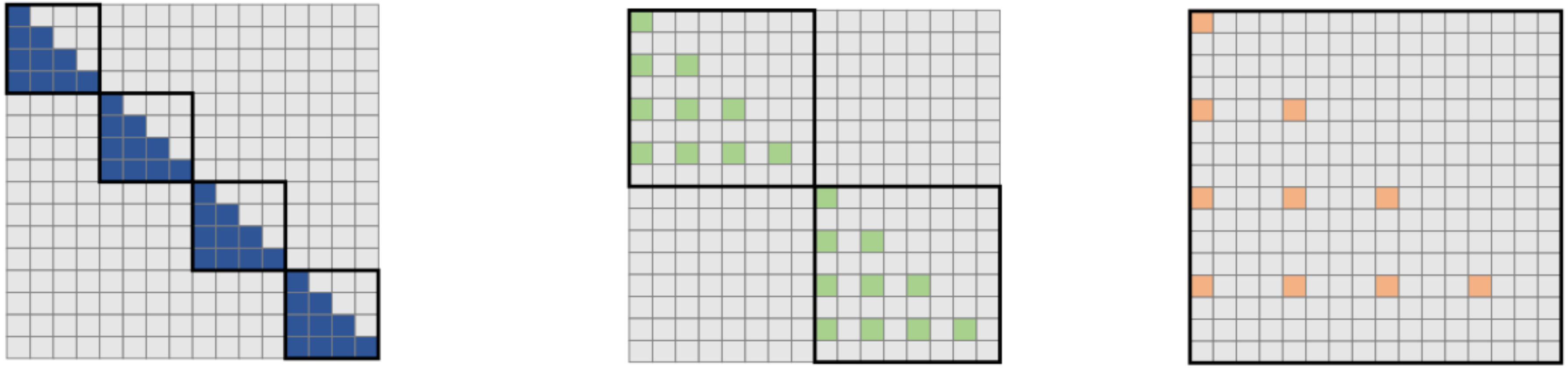
$$\widetilde{K}_i = [K_{iw}, K_{iw+r}, K_{iw+2r}, \ldots, K_{(i+1)w-1}]$$

$$\widetilde{V}_i = [V_{iw}, V_{iw+r}, V_{iw+2r}, \ldots, V_{(i+1)w-1}]$$

$$\widetilde{O}_i = \mathrm{softmax}(\widetilde{Q}_i \widetilde{K}_i^T)\widetilde{V}_i$$

Introduces new parameters *w* and *r*

LongNet: Dilated Attention



w Segment Length: 4
r Dilated Rate: 1

Segment Length: 8
Dilated Rate: 2

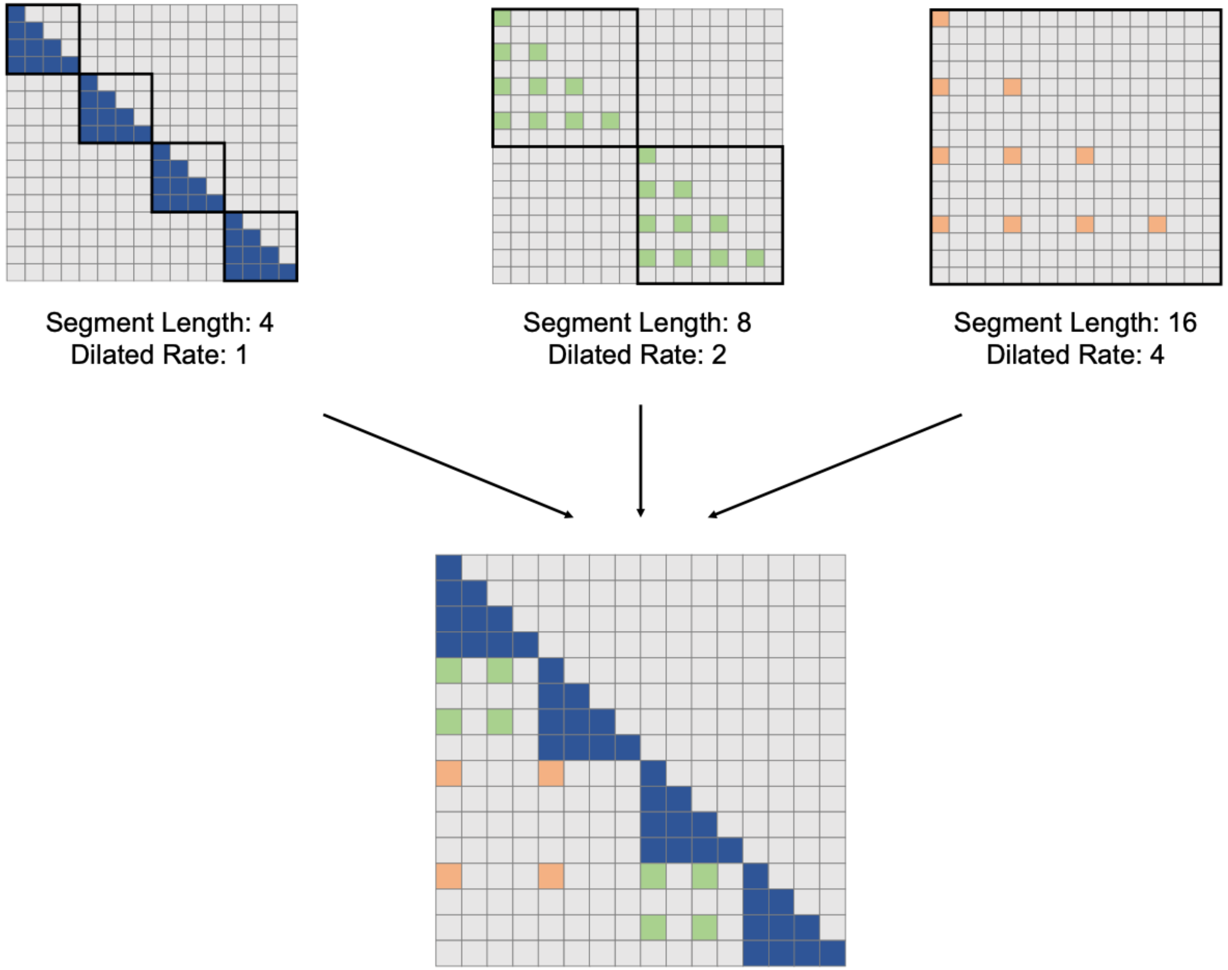
Segment Length: 16
Dilated Rate: 4

Introduces new parameters *w* and *r*

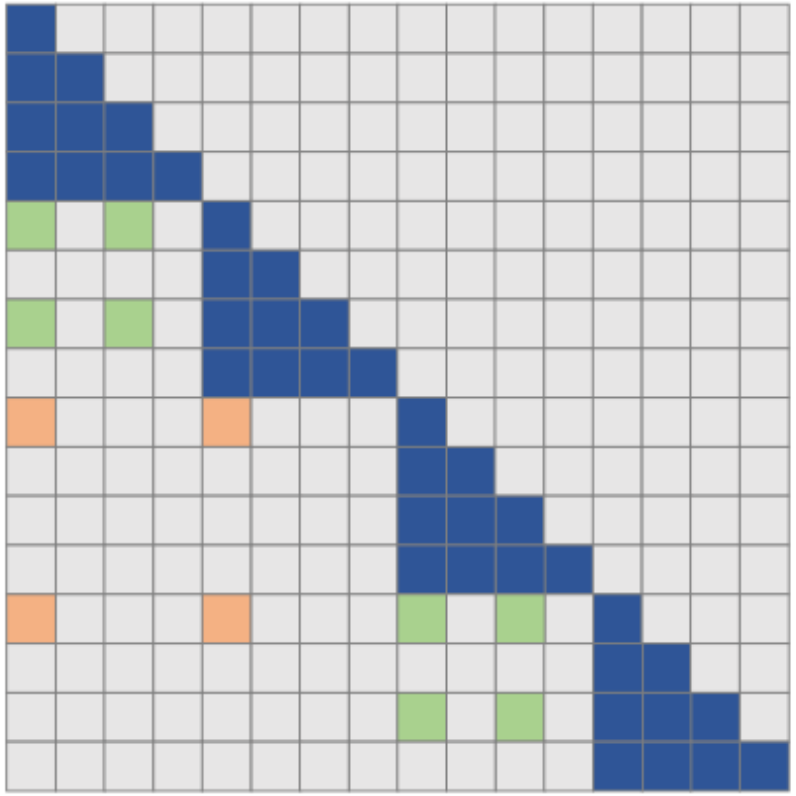
Geometric Sequence: $w = \{w_0, w_1, w_2, \ldots, N\} (w_i < w_{i+1} < N)$

$r = \{1, r_1, r_2, \ldots, r_k\}_k (1 < r_i < r_{i+1})$

LongNet: Dilated Attention



Dilated Attention: Computational Complexity

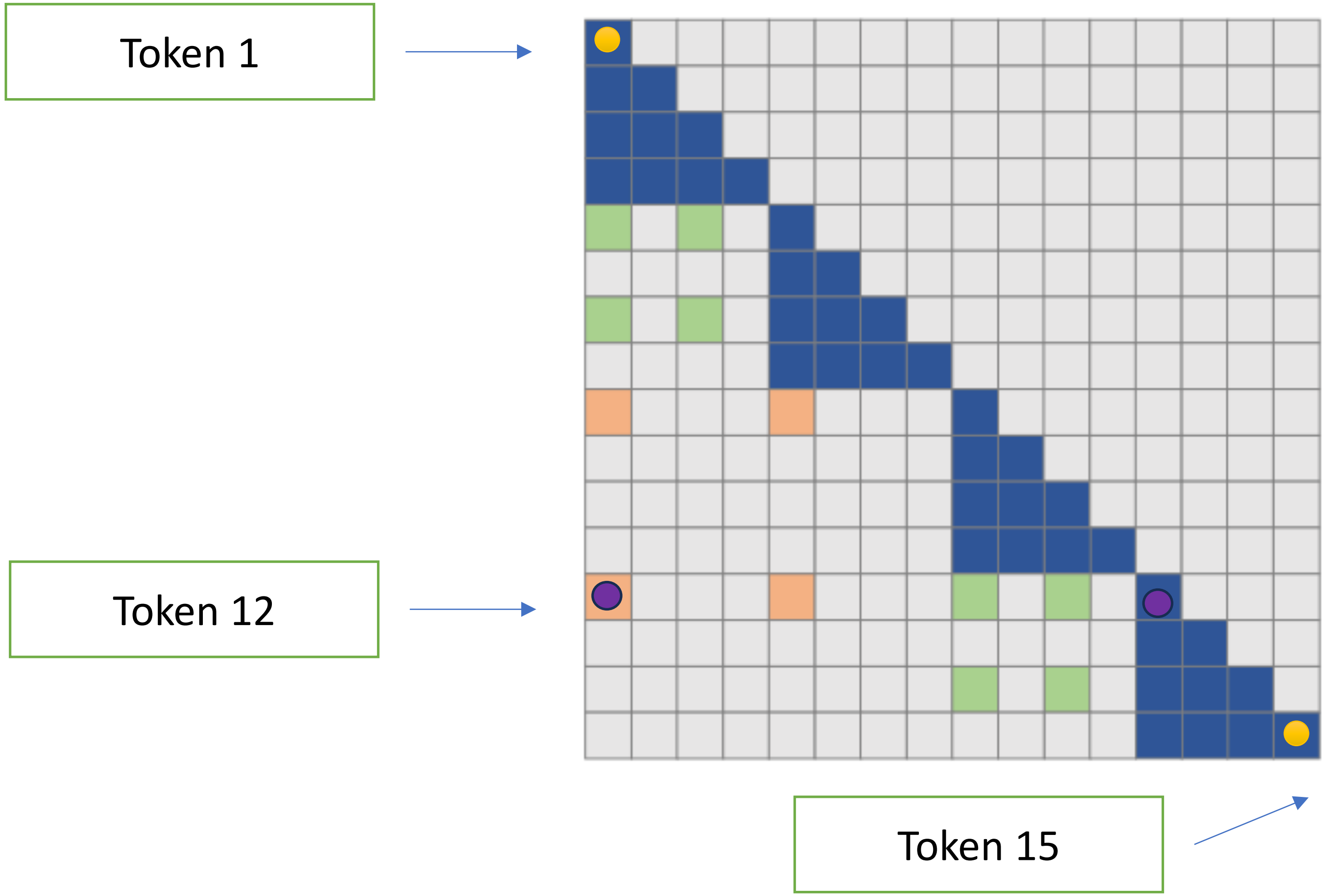


$$FLOPs = \frac{2N}{w}\left(\frac{w}{r}\right)^2 d = \frac{2Nwd}{r^2}$$

$$FLOPs = 2Nd\sum_{i=1}^{k}\frac{w_i}{r_i^2}$$

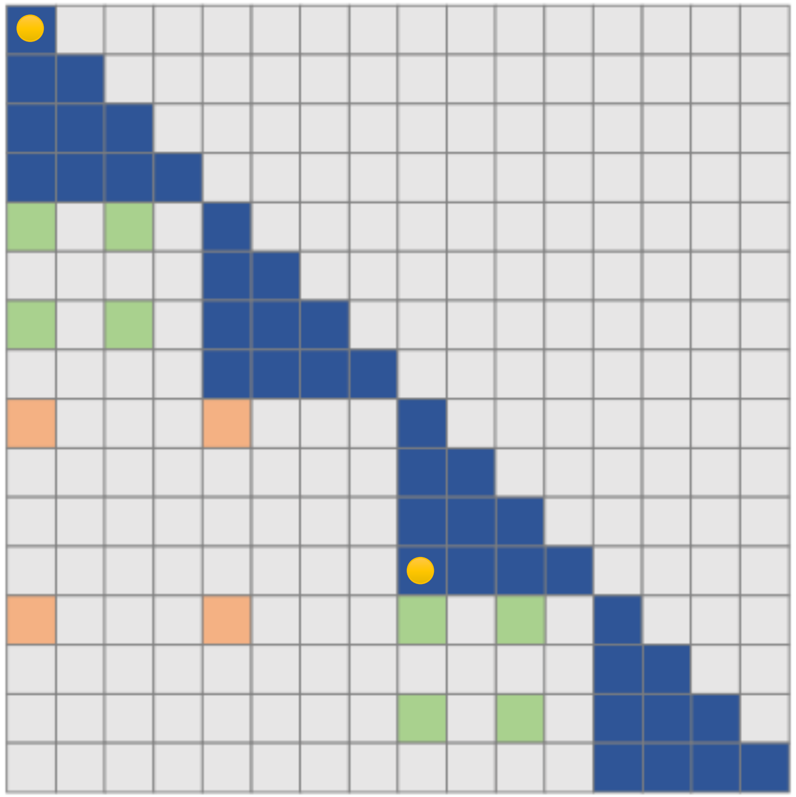
$$\mathcal{O}(Nd)$$

Dilated Attention: Related Attention



Dilated Attention: Related Attention

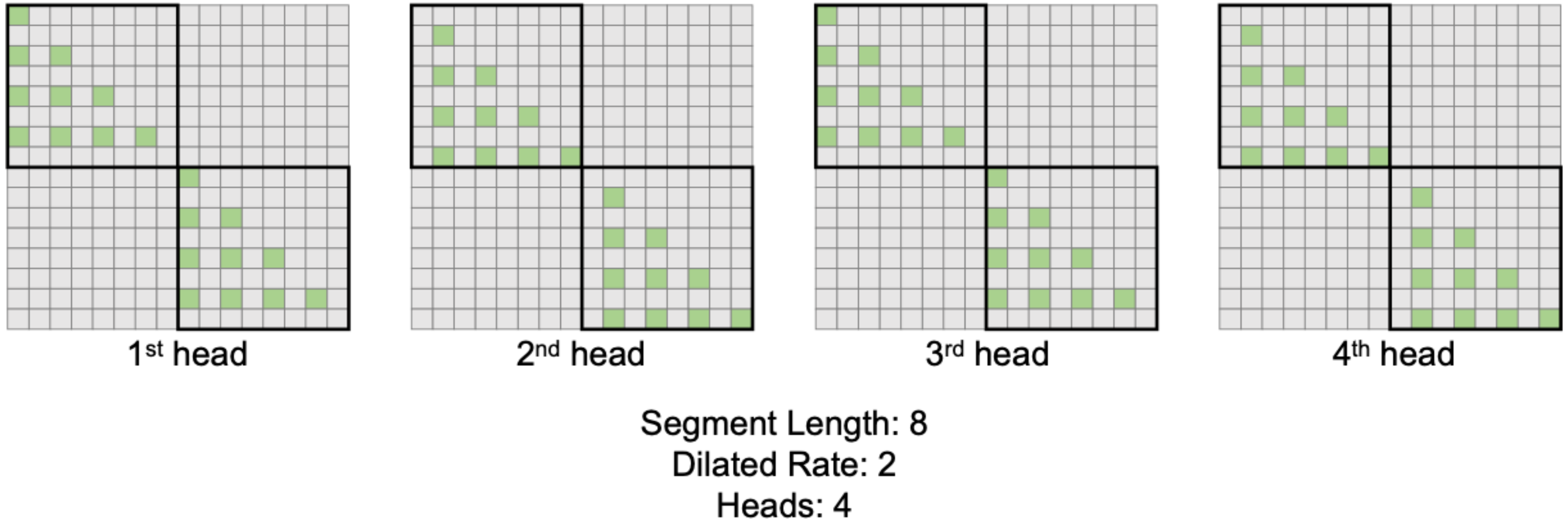
L= $\mathcal{O}(\log N)$



Attention allocation decreases exponentially as the distance between tokens grows

Multihead Attention

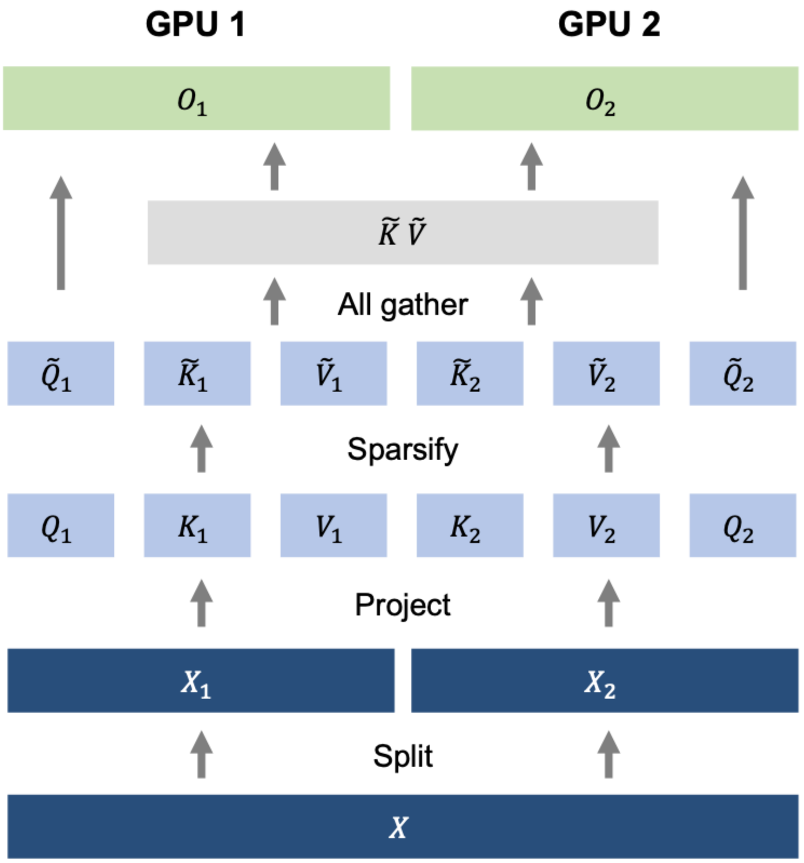
Different computation among different heads sparsify different parts of the query-key-value pairs



Distributed Training

Parallelize training by partitioning the sequence dimension

The computation and communication costs are nearly constant as the number of devices grows

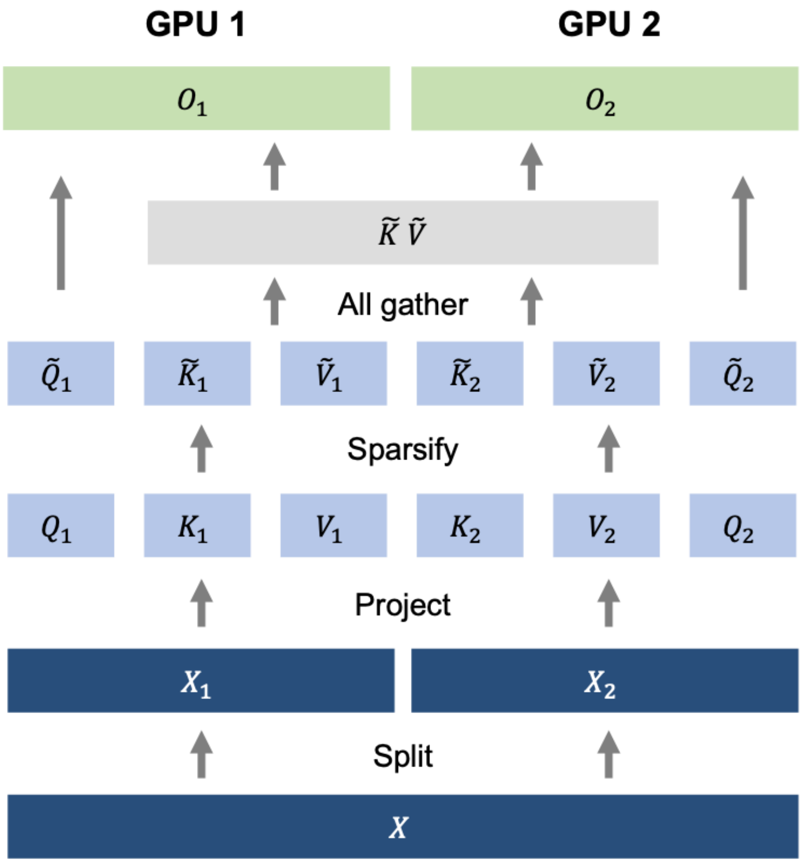


Distributed Training

The concatenation of the outputs across different devices becomes the final attention output:

$$\widetilde{O_1} = \text{softmax}(\widetilde{Q_1}\widetilde{K}^T)\widetilde{V}, \quad \widetilde{O_2} = \text{softmax}(\widetilde{Q_2}\widetilde{K}^T)\widetilde{V}$$

$$\widetilde{O} = [\widetilde{O_1}, \widetilde{O_2}]$$

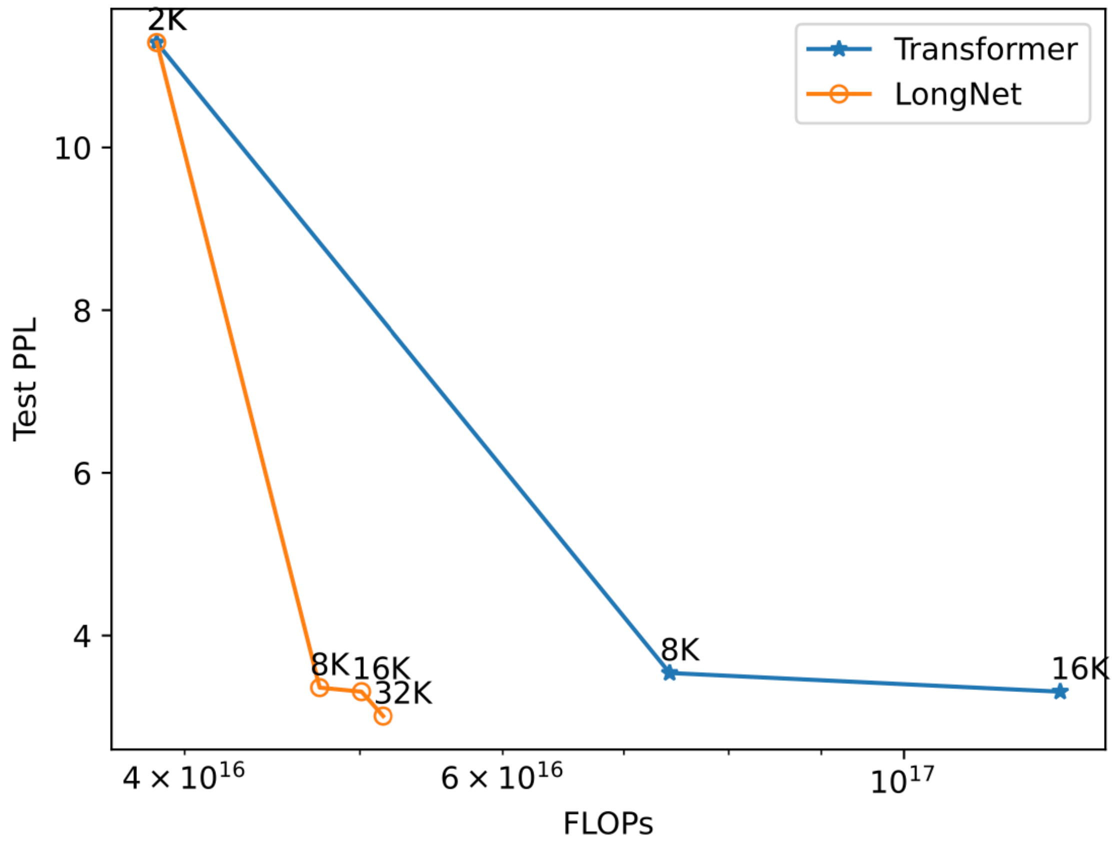


Scaling up to 1B tokens

Increasing the sequence length during training generally leads to a better language model

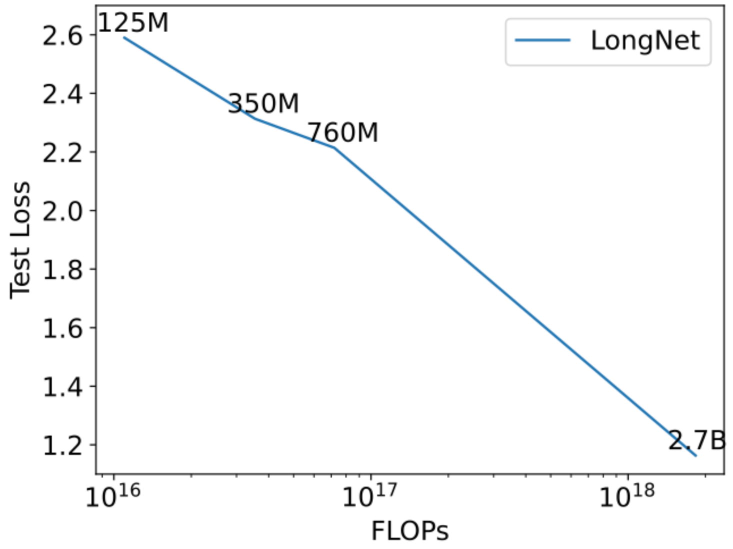
Model	Length	Batch	Github 2K	Github 8K	Github 32K
Transformer [VSP+17]	2K	256	4.24	5.07	11.29
Sparse Transformer [CGRS19]	8K	64	4.39	3.35	8.79
LongNet (ours)			4.23	3.24	3.36
Sparse Transformer [CGRS19]	16K	32	4.85	3.73	19.77
LongNet (ours)			4.27	3.26	3.31
Sparse Transformer [CGRS19]	32K	16	5.15	4.00	3.64
LongNet (ours)			4.37	3.33	3.01

Results

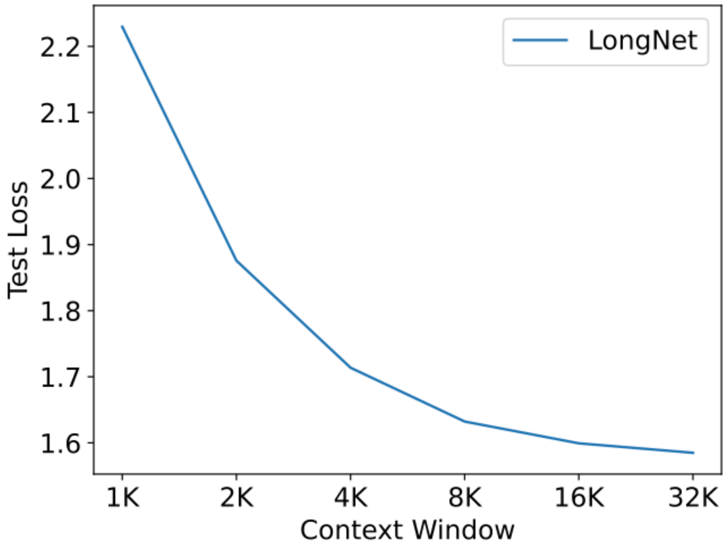

Transformer
LongNet
2K
8K
16K
32K
8K
16K
10
8
6
4
Test PPL
4 × 10^16
6 × 10^16
10^17
FLOPs

Scaling up Model Size

Model Size



Context Window



Dense Transformer is not a prerequisite for scaling the language models

Memorizing Transformers

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, Christian Szegedy

{yuhuai,mrabe,delesley,szegedy}@google.com

2022

Growing Knowledge Base

Theorem database in mathematics

Lemma 3.3 *Let $\varphi \in L^2(\mathbb{K})$ be a refinable function with refinement mask $m_0(\xi)$ such that condition (3.5) is satisfied. Then, $P_j = \sum_{k\in\mathbb{N}_0} |\langle f, \varphi_{j,k}\rangle|^2 < \infty$, for any function $f \in L^2(\mathbb{K})$ and*

$$(i) \quad \lim_{j\to\infty} P_j = \|f\|^2; \qquad (ii) \quad \lim_{j\to-\infty} P_j = 0$$

where $\varphi_{j,k}(x) = q^{j/2}\varphi\big(\mathfrak{p}^{-j}x - u(k)\big), j \in \mathbb{Z}, k \in \mathbb{N}_0.$

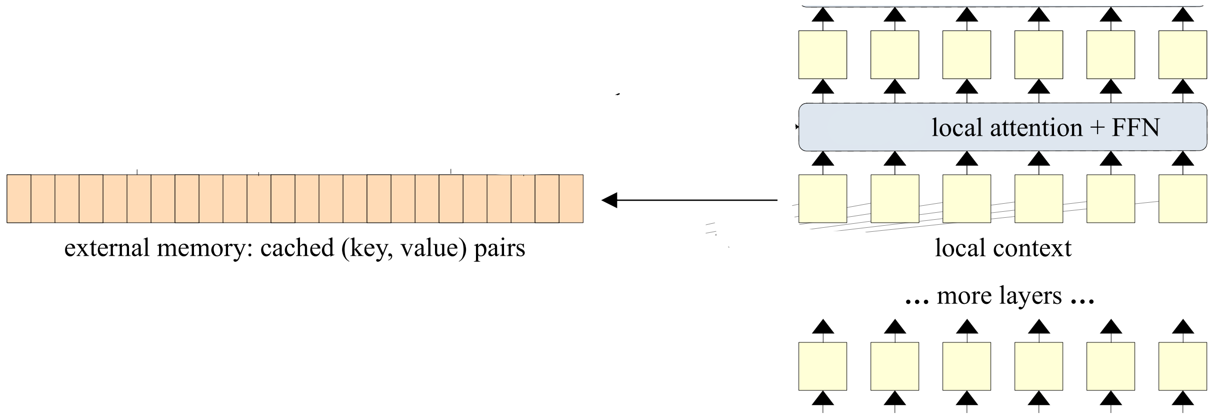
Proof: Implementation of Plancherel and Parseval formulae yields

$$\begin{aligned}
&\sum_{k\in\mathbb{N}_0} |\langle f, \varphi_{j,k}\rangle|^2 \\
&\quad = q^j \sum_{k\in\mathbb{N}_0} \left| \int_{\mathbb{K}} \hat{f}(\xi)\overline{\hat{\varphi}(\mathfrak{p}^j\xi)}\chi_k(\mathfrak{p}^j\xi)\,d\xi \right|^2 \\
&\quad = q^j \sum_{k\in\mathbb{N}_0} \left| \int_{\mathfrak{B}^{-j}} \left[\sum_{n\in\mathbb{N}_0} \hat{f}(\xi + \mathfrak{p}^{-j}u(n))\overline{\hat{\varphi}(\mathfrak{p}^j(\xi + \mathfrak{p}^{-j}u(n)))} \right] \chi_k(\mathfrak{p}^j\xi)\,d\xi \right|^2 \\
&\quad = \int_{\mathfrak{B}^{-j}} \left| \sum_{n\in\mathbb{N}_0} \hat{f}(\xi + \mathfrak{p}^{-j}u(n))\overline{\hat{\varphi}(\mathfrak{p}^j(\xi + \mathfrak{p}^{-j}u(n)))} \right|^2 d\xi \\
&\quad = \|F_j\|^2, \qquad\qquad (3.8)
\end{aligned}$$

Codebase in program synthesis

Memorizing Transformers

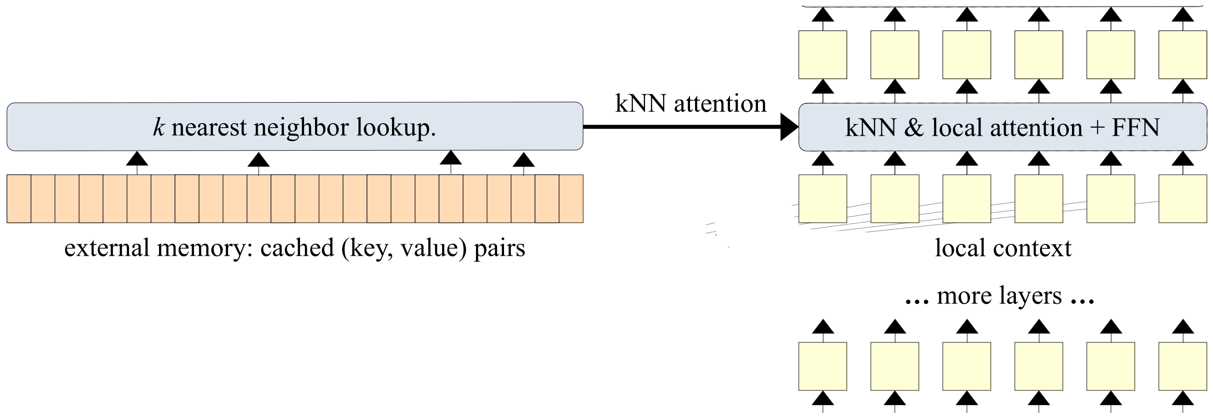
Maintain a memory previously generated keys and values



Memorizing Transformers

Maintain a memory previously generated keys and values

Approximate attention into memory via kNN for scalability

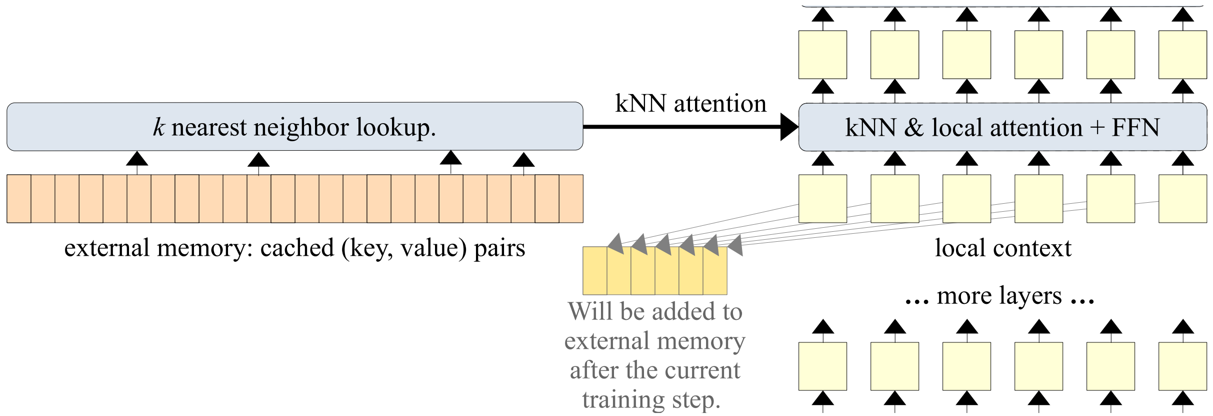


Memorizing Transformers

Maintain a memory previously generated keys and values

Approximate attention into memory via kNN for scalability

Fast maximum inner product search algorithm – O(logN)



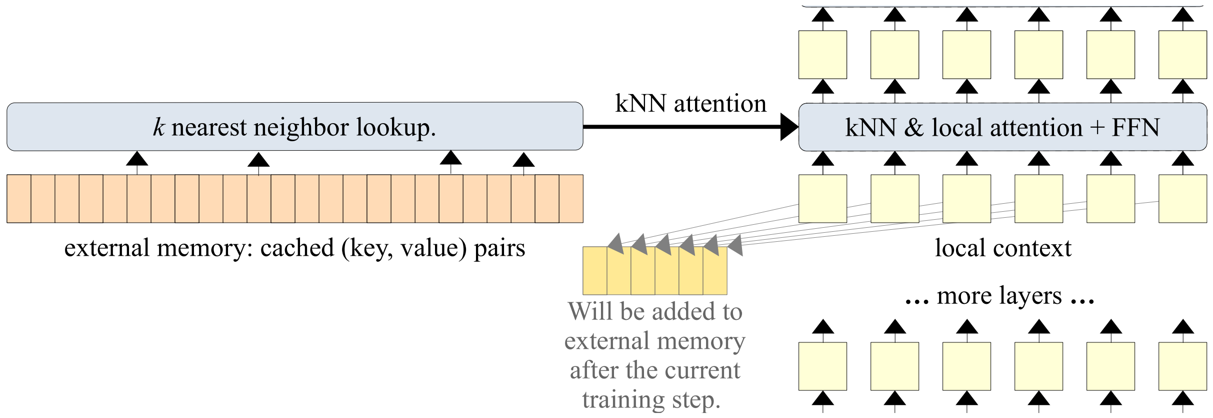
Memorizing Transformers

Maintain a memory previously generated keys and values

Approximate attention into memory via kNN for scalability

Fast maximum inner product search algorithm – O(logN)

Do not backpropagate into memory



Memory update

- Split sequence into subsequences in consecutive order
- Store keys and values in memory after subsequence is read
- If memory is not empty, the current subsequence can attend to the memory

Attending to both context and memory

- Originally only attend to context’s key and value
- Perform a top-k attention using fast maximum inner search into memory
  - Retrieve the top-k keys and value and perform attention on them
- Allows for large scale memory without computational constraint
- This is approximate k-NN search

Attending to both context and memory

$$\boldsymbol{V}_a = \boldsymbol{V}_m \odot g + \boldsymbol{V}_c \odot (1 - g)$$

$\boldsymbol{V}_a$ - Next token combined result of attention

$\boldsymbol{V}_m$ - Attention from external memory

$\boldsymbol{V}_c$ - Attention from context

g – learnable parameter between 0 and 1

Results

Context	Memory	XL cache	arXiv	PG19	C4(4K+)	GitHub	Isabelle
512	None	None	3.29	13.71	17.20	3.05	3.09
2048	None	None	2.69	12.37	14.81	2.22	2.39
512	None	512	2.67	12.34	15.38	2.26	2.46
2048	None	2048	2.42	11.88	14.03	2.10	2.16
512	1536	None	2.61	12.50	14.97	2.20	2.33
512	8192	None	2.49	12.29	14.42	2.09	2.19
512	8192	512	2.37	11.93	14.04	2.03	2.08
512	65K	512	2.31	11.62	14.04	1.87	2.06
2048	8192	2048	2.33	11.84	13.80	1.98	2.06
2048	65K	2048	**2.26**	**11.37**	**13.64**	**1.80**	**1.99**

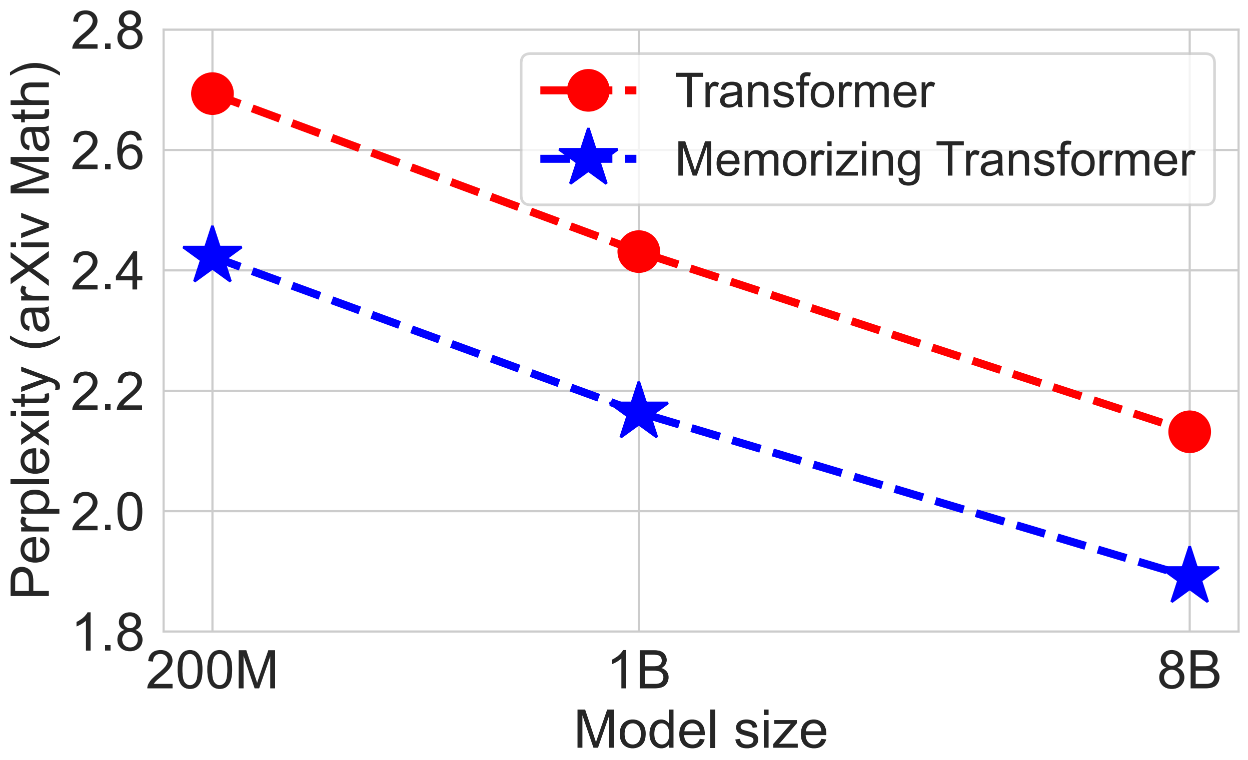
Adding external memory results in substantial gains across datasets and architectures

Increasing the size of the memory increases the benefit of the memory

Scaling Model

- Smaller Memorizing Transformer with just 8k tokens improves perplexity

- 8K Memory attained results comparable to 5-8x larger model



Finetuning transformer to use memory

Finetune 20K steps to obtain 85% of the benefits brought by memory