FlexPrefill: A Context-Aware Sparse Attention Mechanism for Efficient Long-Sequence Inference

Xunhao Lai *[1], Jianqiao Lu *[2], Yao Luo [3], Yiyuan Ma [3], Xun Zhou [3]
[1] Peking University [2] The University of Hong Kong [3] ByteDance Inc
* Leading co-authors with equal contribution

PEKING UNIVERSITY
THE UNIVERSITY OF HONG KONG
ByteDance

Introduction

Long context inference:

Large language models (LLMs) encounter computational challenges during long sequence inference, especially in the attention pre-filling phase, where the complexity grows quadratically with the prompt length.

Our Contribution:

Propose FlexPrefill, a novel, flexible sparse pre-filling attention mechanism designed to adapt in real time to the specific needs of each input and attention head.

Project GitHub

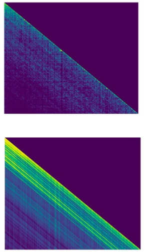
Method Overview

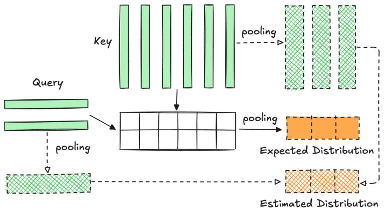
1. Classify attention heads into Query-Aware patterns and Vertical-Slash patterns using JS divergence.
2. Select important query/key blocks until the cumulative attention threshold is reached.
3. Implement a hardware-aligned attention kernel to perform attention only on selected blocks.

Sparse Pattern Determination

Determine with JS divergence:

We classify attention heads into Query-Aware pattern and Vertical-Slash pattern. We take the last few queries to compute the true block-wise score distribution and the estimated distribution. Then, we identify a head as a Query-Aware head when $D_{JS} < \tau$; otherwise, we use the Vertical-Slash pattern.



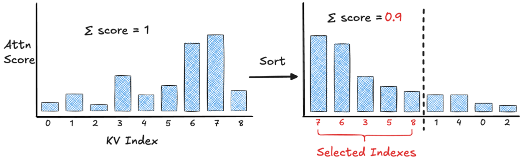


$$D_{JS}(\bar{\boldsymbol{a}},\hat{\boldsymbol{a}}) = \sqrt{JSD(\bar{\boldsymbol{a}}||\hat{\boldsymbol{a}})} = \sqrt{\frac{1}{2}\left(D_{KL}(\bar{\boldsymbol{a}}||\boldsymbol{m}) + D_{KL}(\hat{\boldsymbol{a}}||\boldsymbol{m})\right)}$$

Sparse Index Selection

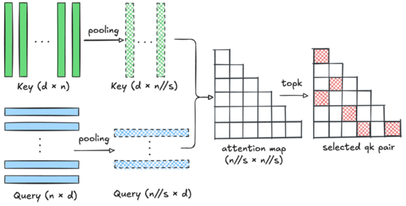
Cumulative attention threshold:

Define a threshold γ for cumulative attention scores. Select tokens with the highest scores until the threshold is reached.



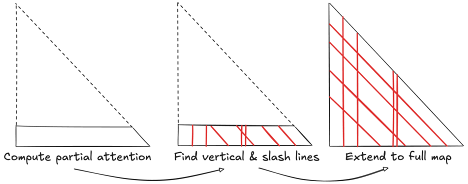
Query-Aware pattern:

Compute attention scores using pooled queries and keys, then sort and select the top blocks until their cumulative score exceeds γ.



Vertical-Slash pattern:

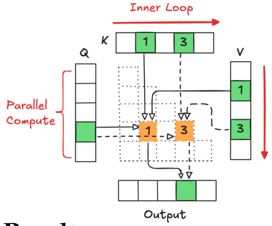
Use last few queries to get partial attention and select vertical and slash lines until their cumulative score exceeds γ. Then extend to the full attention map to get selected QK pairs.

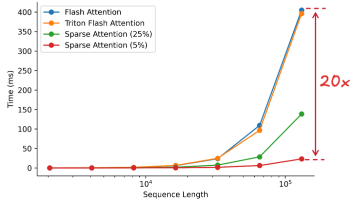


Sparse Attention Calculation

Customized GPU kernel:

Implement sparse attention based on FlashAttention, and only calculate attention over selected KV blocks for each query block.





Results

RULER:

Models	Methods	4k	8k	16k	32k	64k	128k	Avg
LLaMA	Full-attn	95.67	93.75	93.03	87.26	84.37	78.13	88.70
	Streaming LLM	95.43	**93.99**	74.76	48.56	26.20	30.77	61.62
	MInference	**95.67**	**93.99**	93.27	86.54	**84.86**	58.17	85.42
	Ours	95.43	93.51	**94.71**	**89.42**	82.93	**79.09**	**89.18**

InfiniteBench:

Models	Methods	En.Sum	En.QA	En.MC	En.Dia	Zh.QA	Code.Debug	Math.Find	Retr.PassKey	Retr.Number	Retr.KV	Avg
LLaMA	Full-attn	31.91	25.92	69.43	21.50	31.95	16.75	24.29	99.15	99.66	60.00	48.06
	Streaming LLM	30.15	10.15	41.05	8.50	22.38	8.63	17.71	2.71	5.93	0.00	14.72
	Minference	31.04	22.00	63.76	14.50	28.70	5.33	27.43	56.78	77.12	14.00	34.06
	Ours	**31.82**	**24.82**	**69.43**	**19.50**	**35.46**	**16.75**	**31.14**	**98.64**	**99.83**	**44.00**	**47.14**

Attention Speedup:

γ	RULER Score	128k speedup
1 (Full-Attn)	88.70	-
0.97	89.34	1.89x
0.95	89.18	2.43x
0.9	88.38	3.49x
0.85	87.10	4.60x

Performance-Efficiency Balance: