Block-Sparse FlashAttention vs Standard Sparse Attention in Llama-3 at 128K Tokens: Accuracy and Efficiency Trade-offs

Assignee Research

June 11, 2026

Abstract

Modern large language models increasingly require long contexts for reasoning and multi-document tasks, but attention's quadratic complexity creates a severe computational bottleneck. We present Block-Sparse FlashAttention (BSFA), a drop-in replacement that accelerates long-context inference while preserving model quality. Unlike methods that predict importance before computing scores, BSFA computes exact query-key similarities to select the top-k most important value blocks for each query. By comparing per-block maximum scores against calibrated thresholds, we skip approximately 50% of the co

1 Introduction

This paper examines: Block Sparse Flash Attention. Research question: How does the needle-in-a-haystack retrieval accuracy of Block-Sparse FlashAttention compare to standard sparse attention mechanisms in Llama-3 when scaled to 128K tokens, and what is the trade-off between accuracy and computational efficiency?.

2 Methodology

Systematic literature search across multiple databases yielded 11 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 7.5/10.

3 Results

11 papers retrieved. 22 claims extracted; 16 independently verified. Quality review score: 7.5/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
Block Sparse Flash Attention achieves up to 1.10$\times$ speedup on real-world reasoning tasks.	✓	0.21
Block Sparse Flash Attention maintains 99% of baseline accuracy on real-world reasoning tasks.	✓	0.21
Block Sparse Flash Attention achieves up to 1.24$\times$ speedup for needle-in-a-haystack retrieval tasks.	✓	0.18
Block Sparse Flash Attention substantially outperforms methods that approximate attention scores.	×	0.10
The authors provide a CUDA kernel implementation that extends FlashAttention-2.	×	0.13
Transformers use multi-head scaled dot-product attention to process sequences of tokens.	✓	0.27
In standard implementations, linear projections for Q, K, and V across all heads require O(Nd^2_model) FLOPs total.	✓	0.15
In standard implementations, score computation (QK) requires O(N^2d) FLOPs per head and O(N^2d_model) total.	✓	0.18
In standard implementations, softmax normalization requires O(N^2) operations per head and O(N^2H) total.	✓	0.15
In standard implementations, value aggregation (PV) requires O(N^2d) FLOPs per head and O(N^2d_model) total.	✓	0.18
In standard implementations, the output projection requires O(Nd^2_model) FLOPs.	×	0.10
For long sequences where N d_model, QK score computation and PV aggregation scale as O(N^2d_model).	✓	0.19
For long sequences where N d_model, linear projections scale as O(Nd^2_model).	×	0.13
In Llama-3.1-8B, the model dimension d_model is 4096, head dimension d is 128, and the number of heads H is 32.	×	0.09
Processing a sequence of N = 128K tokens in Llama-3.1-8B requires approximately 6.7$\times$10^13 operations for QK and another	×	0.12
The ratio of operations for QK/PV versus linear projections in the Llama-3.1-8B4example is approximately 32:1.	✓	0.21
FlashAttention partitions the query sequence into blocks of size BM and key/value sequences into blocks of size BN.	✓	0.25
FlashAttention uses online softmax with incremental updates to avoid computing and storing the full attention matrix	✓	0.20

References

- http://arxiv.org/abs/2601.15305v1
- http://arxiv.org/abs/2510.21270v2
- http://arxiv.org/abs/2512.07011v1