Comparative Analysis of Gated Sparse Attention and Block-Sparse FlashAttention for Stability and Accuracy on LongBench with Over

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 Gated Sparse Attention compare to Block-Sparse FlashAttention in terms of training stability and accuracy on the LongBench benchmark when processing documents exceeding 100k tokens?.

2 Methodology

Systematic literature search across multiple databases yielded 7 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 7.3/10.

3 Results

7 papers retrieved. 17 claims extracted; 12 independently verified. Quality review score: 7.3/10.

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.	✓	0.20
Block Sparse Flash Attention maintains 99% of baseline accuracy on real-world reasoning tasks.	✓	0.20
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_model^2) FLOPs.	×	0.15
In standard implementations, score computation (QK) requires O(N^2 * d_model) FLOPs total.	×	0.14
In standard implementations, value aggregation (PV) requires O(N^2 * d_model) FLOPs total.	×	0.14
For Llama-3.1-8B with d_model = 4096, d = 128, and H = 32, processing a sequence of N = 128K tokens requires approximate	✓	0.17
For Llama-3.1-8B with d_model = 4096, d = 128, and H = 32, processing a sequence of N = 128K tokens requires approximate	✓	0.18
For Llama-3.1-8B with d_model = 4096, the ratio of operations between quadratic attention components and linear projecti	✓	0.16
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 storing the full attention matrix.	✓	0.18
Block-Sparse FlashAttention (BSFA) computes all query-key scores exactly to determine importance before deciding which v	✓	0.22
BSFA skips loading and processing value blocks whose maximum scores fall below calibrated thresholds.	✓	0.21
BSFA exploits the observation that blocks with uniformly low scores contribute negligibly after softmax normalization.	✓	0.22

References

- http://arxiv.org/abs/2509.07120v2
- http://arxiv.org/abs/2512.07011v1
- http://arxiv.org/abs/2402.01476v2