Spatial-Temporal Smoothing in ReST-KV vs. Retention Methods on LongBench Memory-Accuracy Trade-offs

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 8 peer-reviewed papers addressing the following research question: How does the spatial-temporal smoothing mechanism in ReST-KV compare to sliding window or top-K retention methods in terms of memory-accuracy trade-offs on LongBench at 256K context lengths. 15 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.7/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: ReST-KV: Robust KV Cache Eviction with Layer-wise Output Reconstruction and Spatial-Temporal Smoothing. Research question: How does the spatial-temporal smoothing mechanism in ReST-KV compare to sliding window or top-K retention methods in terms of memory-accuracy trade-offs on LongBench at 256K context lengths?.

2 Methodology

Systematic literature search across multiple databases yielded 8 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.7/10.

3 Results

8 papers retrieved. 15 claims extracted; 0 independently verified. Quality review score: 3.7/10.

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
ReST-KV is evaluated on five open-source LLMs: Llama2-Chat, Gemma-Instruct, Llama3-Instruct, Mistral-Instruct-v0.3, and	×	0.03
ReST-KV is compared with five baseline methods: StreamingLLM, H2O, TOVA, SnapKV, and LaCache.	×	0.04
ReST-KV is evaluated on four benchmarks: LongBench, RULER, Needle-in-a-Haystack, and InniteBench.	×	0.10
ReST-KV achieves the best performance in most cases on the LongBench benchmark.	×	0.06
ReST-KV reduces peak memory usage by approximately 36.0% compared to full cache at a context length of 128k.	×	0.05
ReST-KV achieves an approximate 10.61$\times$ speedup over the full cache method at a 128K context length.	×	0.11
ReST-KV is compatible with prell sparse attention approaches, yielding a Time-To-First-Token (TTFT) speedup of up to 3.	×	0.09
ReST-KV uses a xed cache budget to limit the number of KV pairs, overcoming the latency bottleneck for long sequences.	×	0.09
ReST-KV only requires computing attention outputs within a small query window, resulting in a computational complexity c	×	0.05
LLMs typically decode text in an auto-regressive manner, which allows them to generate high-quality, contextually cohere	×	0.03
KV cache reduces redundant computation by storing previously computed keys and values.	×	0.04
At each decoding step t, the KV cache stores previously computed keys and values K1:t-1, V1:t-1 for X[1 : t -1].	×	0.05
The model requires only the current token xt to generate xt+1, rather than the full sequence X = [x1, . . . , xt].	×	0.02
The query qt, key kt, and value vt are computed as qt = xtWQ, kt = xtWK, vt = xtWV.	×	0.02
The currently computed kt and vt will be concatenated with the previously cached keys and values, and used in the attent	×	0.04

References

- http://arxiv.org/abs/2308.14508v2
- http://arxiv.org/abs/2601.02872v1
- http://arxiv.org/abs/2605.08840v1