LiteCache GPU-Centric vs CPU-Centric KV Offloading in Batched LLM Inference

Assignee Research

June 8, 2026

Abstract

This report synthesises findings from 12 peer-reviewed papers addressing the following research question: What is the difference in end-to-end inference throughput and CUDA Graph compatibility between LiteCache's GPU-centric management and CPU-centric KV offloading strategies during batched LLM inference. 10 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: Comparative Characterization of KV Cache Management Strategies for LLM Inference. Research question: What is the difference in end-to-end inference throughput and CUDA Graph compatibility between LiteCache's GPU-centric management and CPU-centric KV offloading strategies during batched LLM inference?.

2 Methodology

Systematic literature search across multiple databases yielded 12 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

12 papers retrieved. 10 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
KV cache memory requirements can quickly surpass that of model parameters, especially when handling batches of requests	×	0.08
Some frameworks have tried optimizing memory allocation to reduce internal fragmentation during inference.	×	0.03
Techniques focus on addressing the growth of KV cache memory, applying hierarchical placement of KV cache across GPU and	×	0.09
Attention sparsification is important to identify, prioritize and cache only important KV cache entries.	×	0.05
Techniques alleviate the computation complexity caused by long-context lengths during inference such as KV cache quantiz	×	0.07
The success of KV cache management depends on various factors, including available GPU memory, the LLM size and the requ	×	0.07
We quantify the accuracy impact of KV cache sparsification across standard benchmarks and a custom retention task, ident	×	0.05
This work provides a comparative evaluation and analysis across three distinct KV cache management paradigms.	×	0.09
Modern LLMs are built by stacking dozens to hundreds of Transformer layers, each comprising a multi-head self-attention	×	0.03
Transformer LLM inference consists of two distinct stages: in the prefill stage, the model processes the input prompt of	×	0.07

References

- http://arxiv.org/abs/2511.14510v2
- http://arxiv.org/abs/2604.05012v1
- http://arxiv.org/abs/2511.17593v1