CAKE Adaptive Eviction vs Static Policies in Mixed-Language Code Generation

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 15 peer-reviewed papers addressing the following research question: How does CAKE's adaptive eviction policy compare to static policies (LRU, FIFO) in terms of HumanEval+ pass@1 scores when applied to mixed-language code generation tasks (Python, Java, C++) under. 9 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: CAKE: Cascading and Adaptive KV Cache Eviction with Layer Preferences. Research question: How does CAKE's adaptive eviction policy compare to static policies (LRU, FIFO) in terms of HumanEval+ pass@1 scores when applied to mixed-language code generation tasks (Python, Java, C++) under constrained memory conditions?.

2 Methodology

Systematic literature search across multiple databases yielded 15 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.5/10.

3 Results

15 papers retrieved. 9 claims extracted; 0 independently verified. Quality review score: 3.5/10.

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
CAKE achieves an approximate 48.63% reduction in peak memory usage compared to the full cache implementation with a 128K	×	0.07
CAKE achieves over 10$\times$ speedup in decoding latency compared to the full cache approach when processing sequences with 12	×	0.15
Decoding latency for the full cache method grows significantly as input length increases due to computational demands an	×	0.04
CAKE maintains a relatively stable decoding speed by preserving a fixed amount of KV cache.	×	0.11
When equipped with the CAKE allocation strategy, H2O and SnapKV consistently improve performance across nearly all tasks	×	0.07
The compatibility experiments were conducted on LongBench datasets using Llama2-7B-Chat under Btotal of 128L and 512L.	×	0.03
The preference metric P is defined as H^(1/τ1) * V^(1/τ2), where H represents spatial dispersion and V represents tempor	×	0.04
The preference metric calculation focuses on the submatrix A[-Sw :, : -Sw] of the attention weights, representing a rece	×	0.02
Layers with a high preference score P benefit more from larger KV cache to maintain performance.	×	0.08

References

- http://arxiv.org/abs/2507.05269v3

- http://arxiv.org/abs/2503.12491v2
- http://arxiv.org/abs/2605.09649v1