THE UNIVERSITY of EDINBURGH

ContextPilot

Fast Long-Context Inference via **Context Reuse**

Yinsicheng Jiang*, Yeqi Huang*, Liang Cheng, Cheng Deng, Xuan Sun, Luo Mai

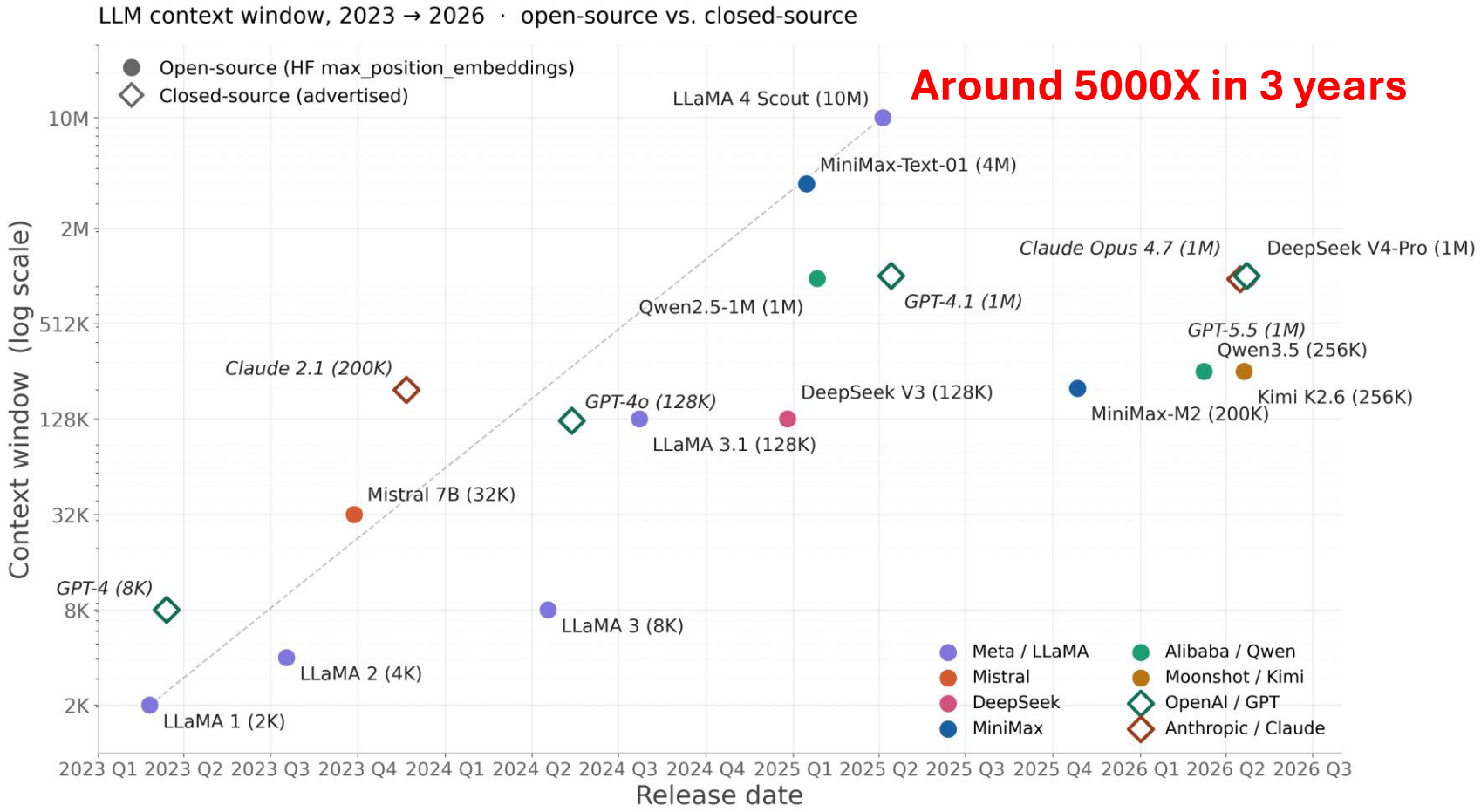
University of Edinburgh

MLSys 2026

github.com/EfficientContext/ContextPilot

ContextPilot

The Rise of Long-Context LLM

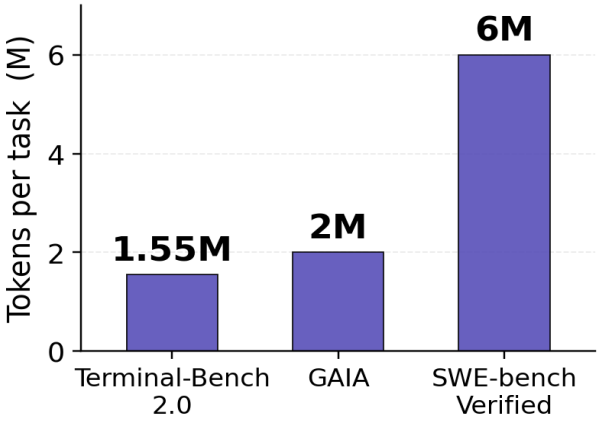


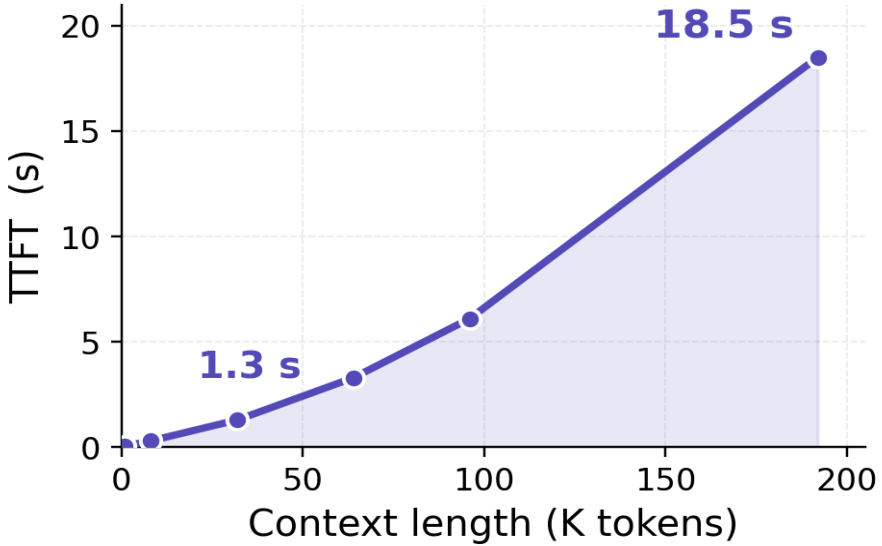
AI Agents Rely on Long Context

OPENCLAW

HERMES AGENT

Drives





Agents accumulate **millions of tokens per task** across several model calls. [1, 2]

Prefill latency scales **superlinearly** with context length. [3]

[1] HAL: A Holistic Agent Leaderboard for Centralized and Reproducible Agent Evaluation– ICLR 2026;

[2] Terminal-Bench 2.0 – ICLR 2026;

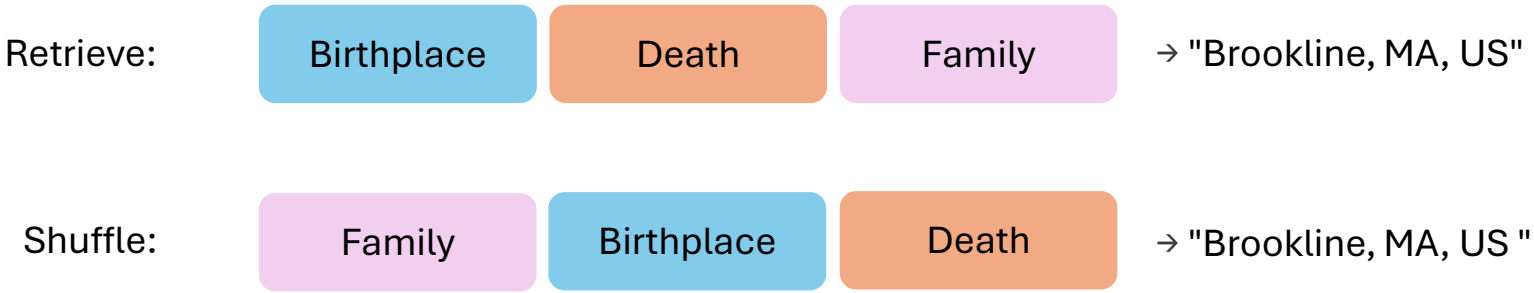
[3] MiniMax M2.5: https://www.minimax.io/news/minimax-m25

Existing Approaches:

We aim to achieve high reuse without sacrificing accuracy.

Key Observation 1: LLMs Are Robust to Context Order

"Where was Kennedy born?"



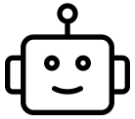
Reproduce [1] with frontier LLMs like GPT-5.2 further confirm this observation.

[1] What Makes a Good Order of Examples in In-context Learning – ACL 2024

Key Observation 2: Contexts Overlap Heavily

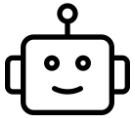
Multi-turn:

"How was Kennedy killed?"

Retrieve → Assassination Info + Death date & place → 

“Killed by gunshot on Nov 12, 1963”

"How was his family after assassination?"

Retrieve → Family Info + Assassination Info → 

“Devastated by grief”

40% doc overlap across turns (MT-RAG)

Multi-session:

"How was Kennedy killed?"

Retrieve → Assassination Info | Death date

"How was Kennedy's family after assassination?"

Retrieve → Family Info | Assassination Info

Prefix cache hit

Context Overlap Rate:

79.2% of MultihopRAG

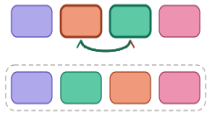
57.4% of NarrativeQA

49.6% of QASPER

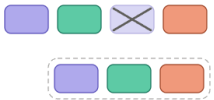
ContextPilot: Context Reuse

Context Reuse: Catch more prefix cache hits, and avoid resending context the LLM has already seen without sacrificing accuracy.

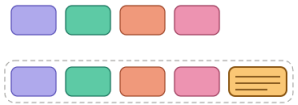
Three Key Mechanisms



1. Context Alignment

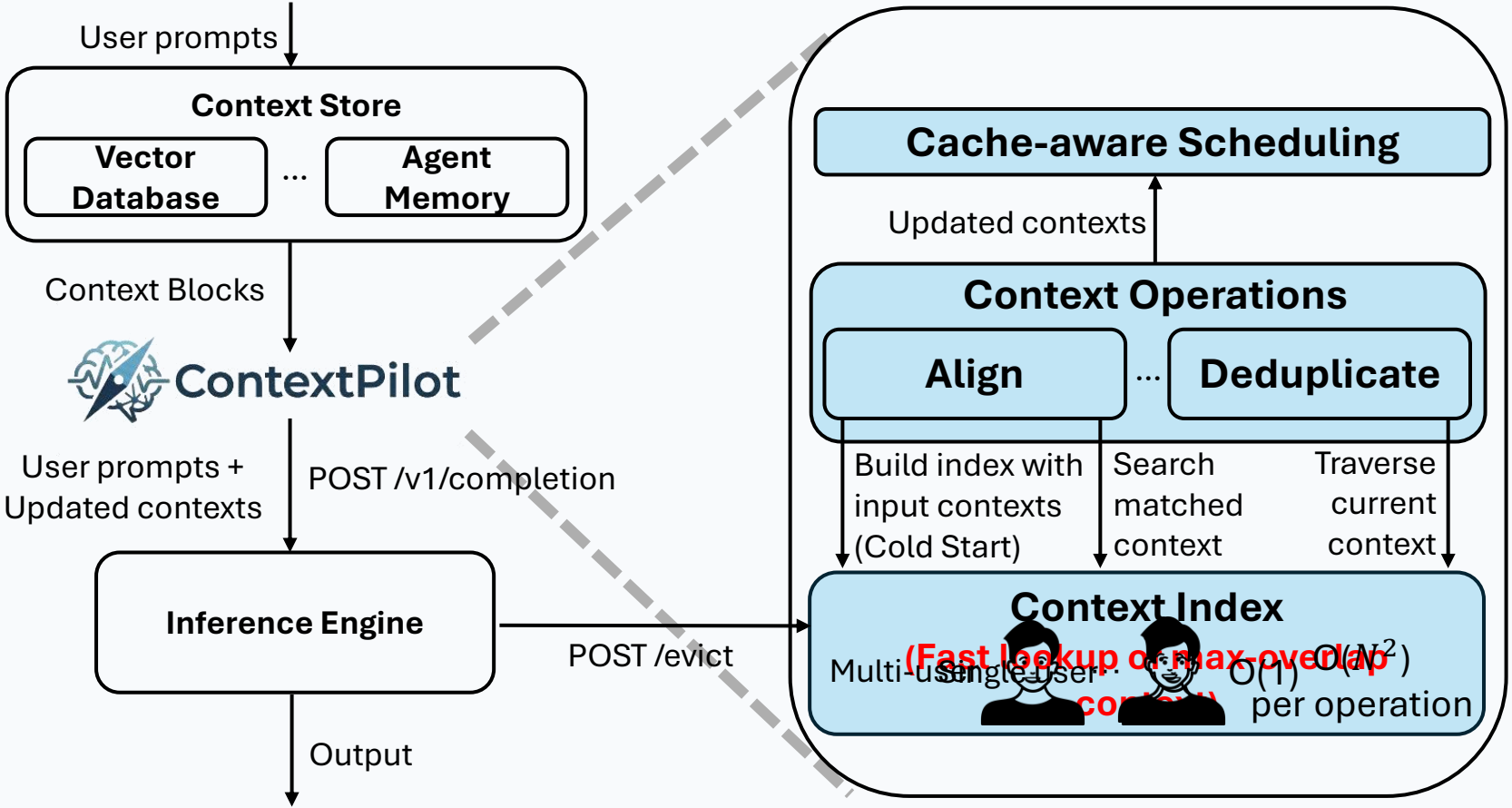


2. Context De-duplication



3. Context Annotations

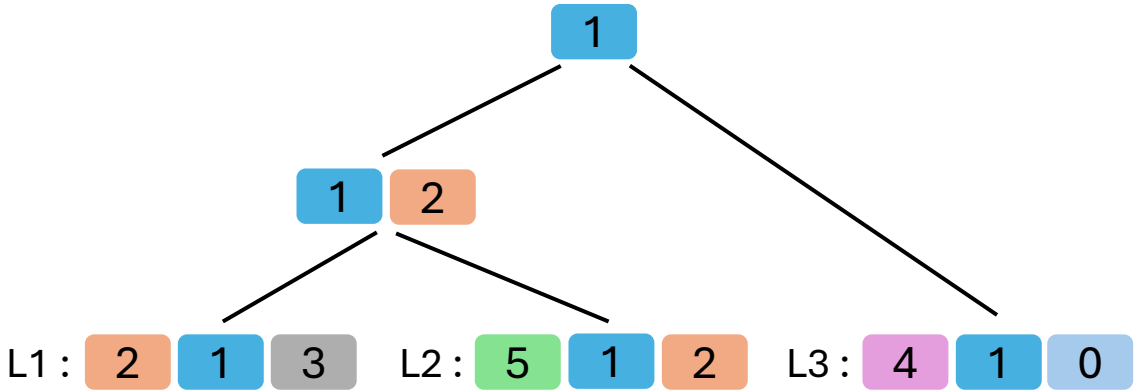
ContextPilot: System Architecture



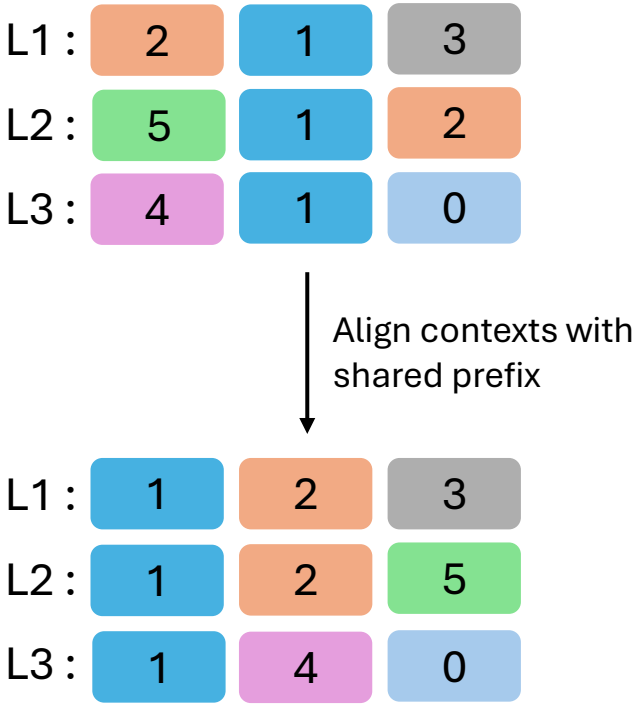
Contribution 1: Context Alignment

Context Index:

Mirror Radix Tree;
Parent: shared prefix of children;
Leaf: full contexts;



Context Alignment:

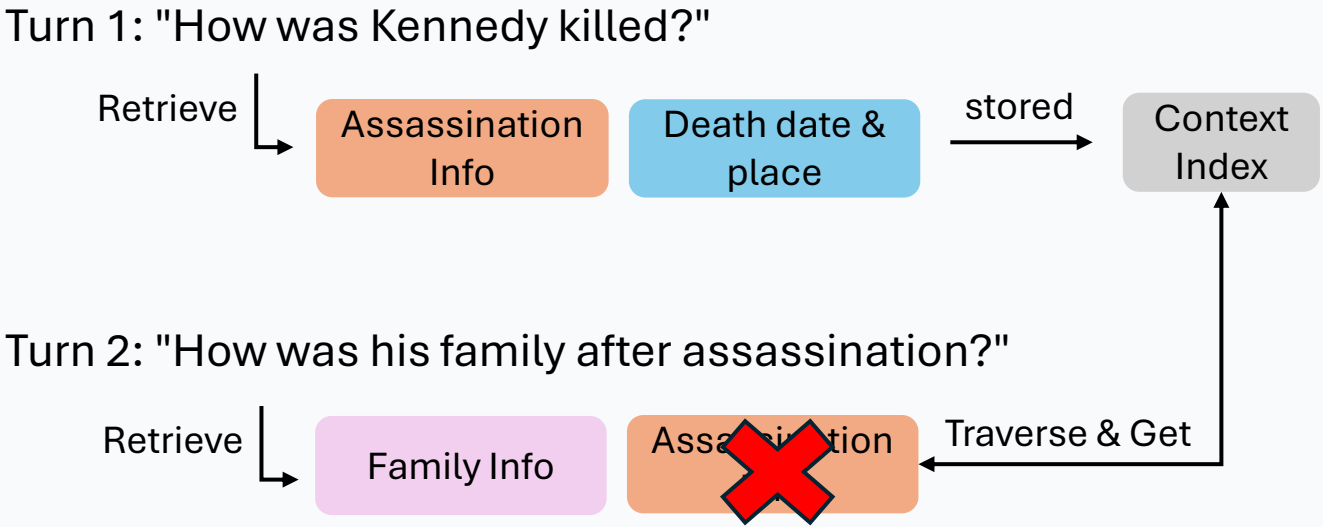


Distance Function:

$$d_{ij} = 1 - \underbrace{\frac{|S_{ij}|}{\max(|C_i|, |C_j|)}}_{\text{Overlap rate}} + \alpha \cdot \underbrace{\frac{\sum_{k \in S_{ij}} |p_i(k) - p_j(k)|}{|S_{ij}|}}_{\text{Context Ranking Distance}}$$

Before: 5% Cache Hit (System Prompt)
After: 33%-67% Cache Hit

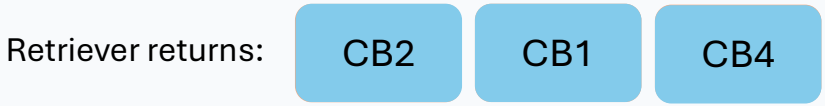
Context De-duplication



Contribution 3: Context Annotations

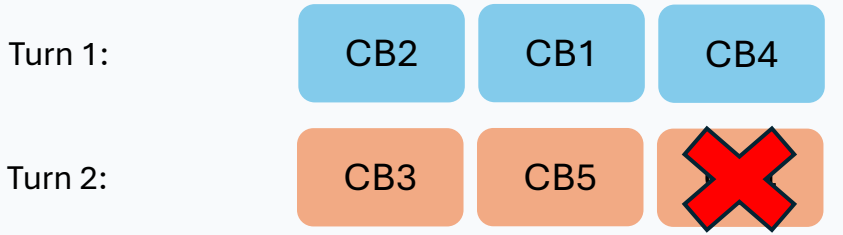
Accuracy loss from alignment and deduplication is about 0.1–3.3%.

Order Annotation (for Alignment)



"Please read in priority order: [CB2] > [CB1] > [CB4]"

Location Annotation (for Deduplication)

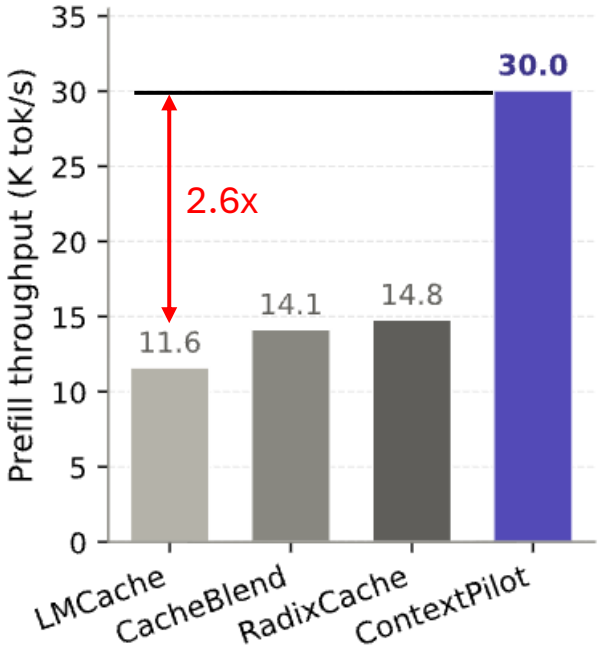


"Refer to [CB4] in Turn 1."

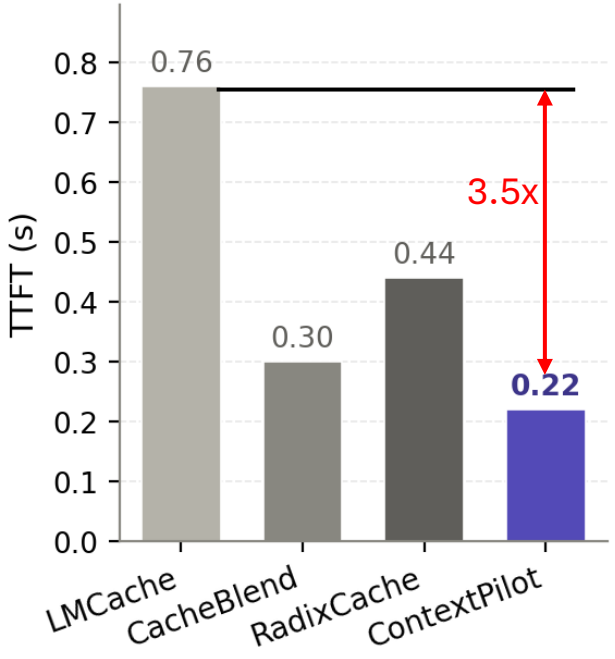
Negligible token overhead, recovers lost accuracy

Results: RAG Workloads

Offline large batch RAG (Llama3.3-70B):



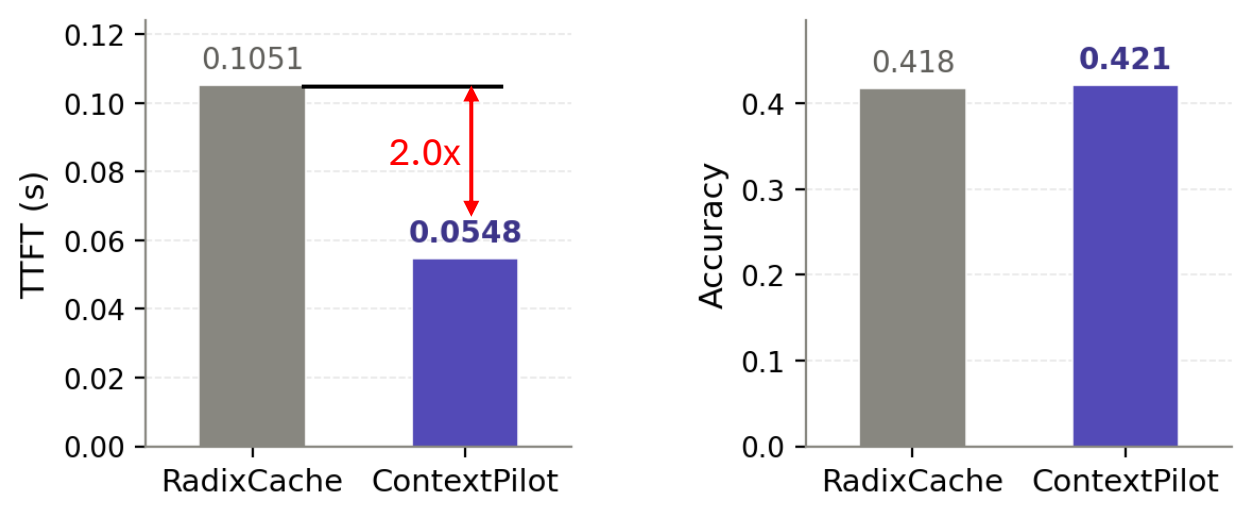
Online multi-turn RAG (Qwen3-4B):



Setup: H100-SXM; SGLang v0.4.6; LMCache: v0.3.8

Agentic Memory (Mem0) + ContextPilot

ContextPilot achieves **2.0x TTFT speedup** without accuracy loss;



RTX A6000; LoCoMo; Qwen3-4B; SGLang v0.5.9

OpenClaw + ContextPilot

ContextPilot achieves **1.8-3.8x TTFT speedup** on both workloads;

Metrics	Method	Document Analysis		Coding	
		Avg	P99	Avg	P99
TTFT (s)	RadixCache	7.2	25.4	5.8	8.6
	ContextPilot	2.6	6.7	2.2	4.9
			3.8x		1.8x

RTX 5090; Claw-Tasks; Qwen3-4B; SGLang v0.5.9

Minimal Overhead, Robust Gains

Per-Request Overhead

~0.7 ms

total (search + alignment + dedup)

Much smaller than prefill latency

Accuracy with Annotations

+1.4 to +4.4%

over aligned-only baseline

Annotation can successfully call back accuracy loss of alignment

ContextPilot: Key Takeaways

- 3.5x prefill speedup with preserved accuracy

ContextPilot Support Matrix			
Inference Engine	vLLM	SGLang	Llama.cpp
RAG & Memory	PageIndex		Mem0
Agent Frameworks	OpenClaw	Hermes Agent	OpenCode

ContextPilot is actively maintained and welcomes community contributions.

New Features: Catch and remove repeated content across context blocks

What's next: Knowledge-aware cache reuse; Broader agent framework support: NanoClaw, Codex, ...

github.com/EfficientContext/ContextPilot

openclaw plugins install @contextpilot-ai/contextpilot

hermes plugins install EfficientContext/ContextPilot

Contact:
ysc.jiang@ed.ac.uk

ContextPilot