THE UNIVERSITY of EDINBURGH

ContextPilot

Fast Long-Context Inference via Context Reuse

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

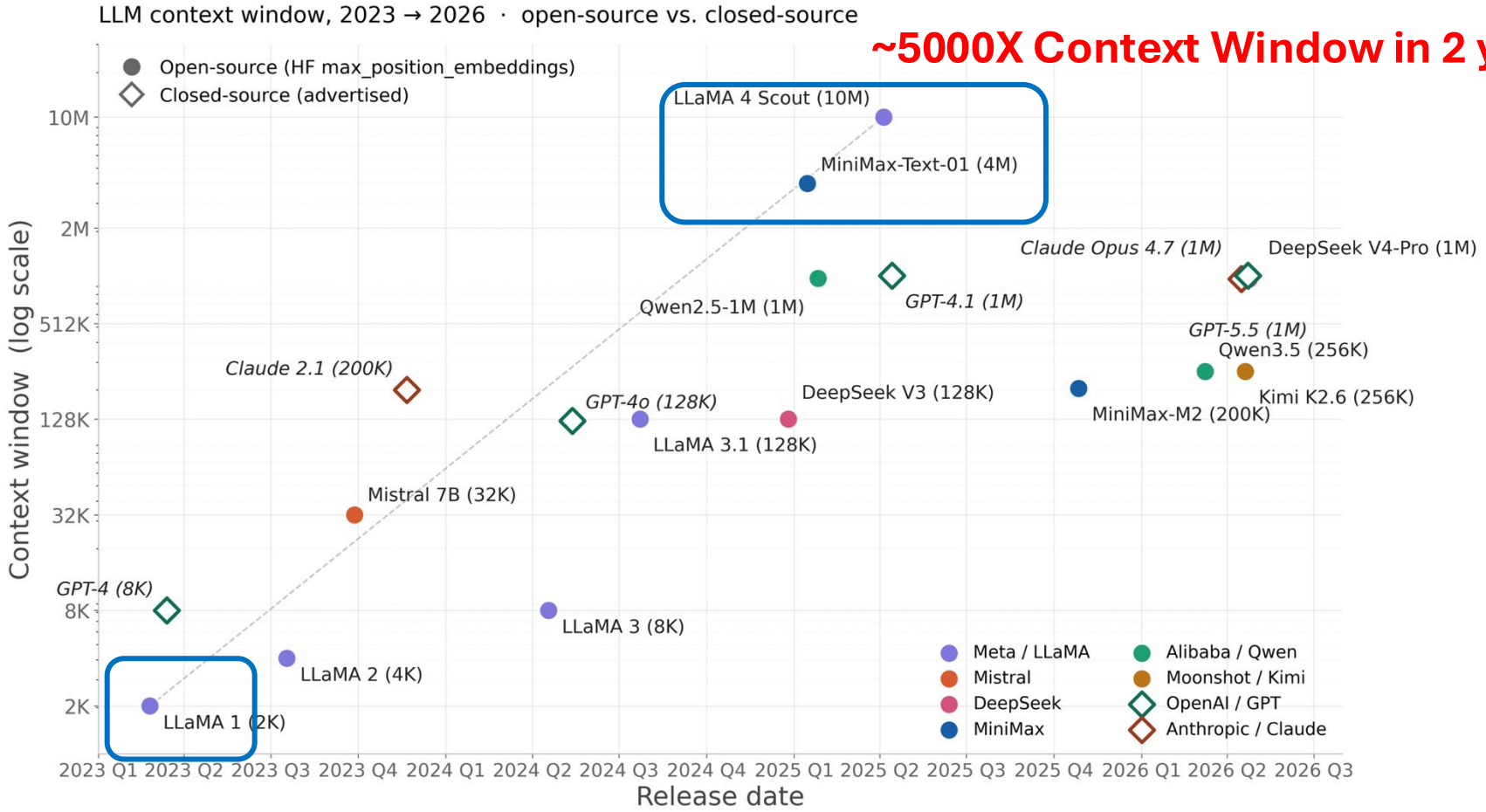
University of Edinburgh

MLSys 2026

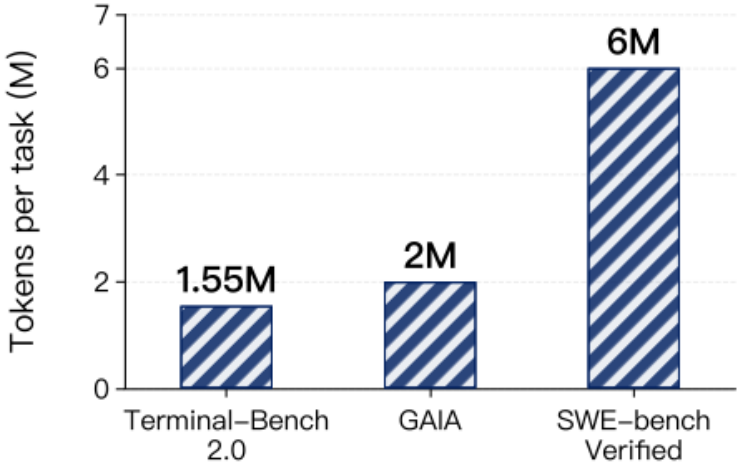
github.com/EfficientContext/ContextPilot

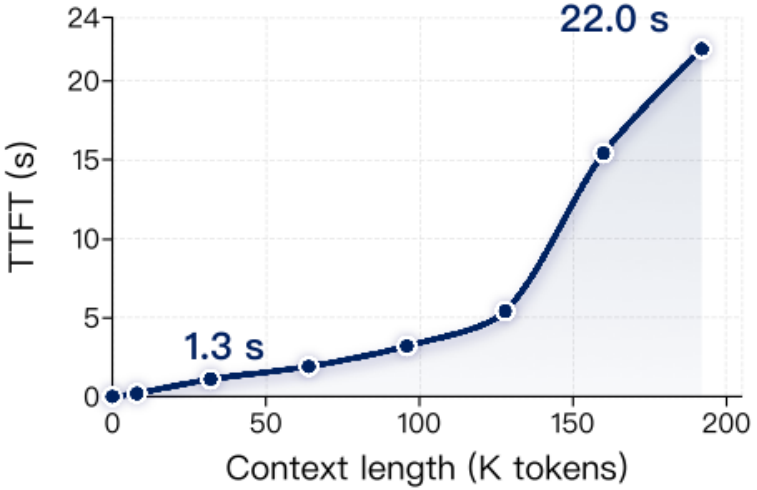
ContextPilot

LLM Context Window Keeps Growing



As Context Grows, Prefill Becomes The Bottleneck





Prefill stage is growing heavier. [1, 2]

- Multi-turn (long history session)
- Agentic application (huge skill docs)
- Coding agent (enormous code base)

TTFT 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

Prefix Cache Can Improve Inference Efficiency

Prompts with Context Blocks / Chunks

Req 1

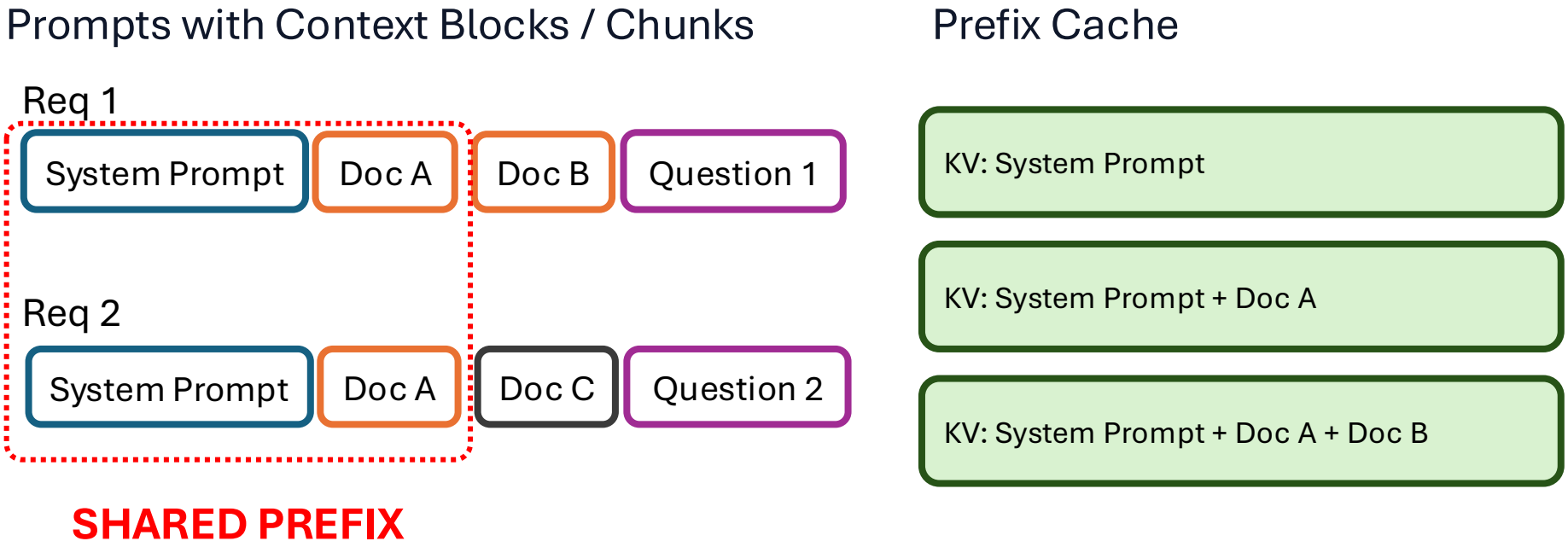
Prefix Cache

KV: System Prompt

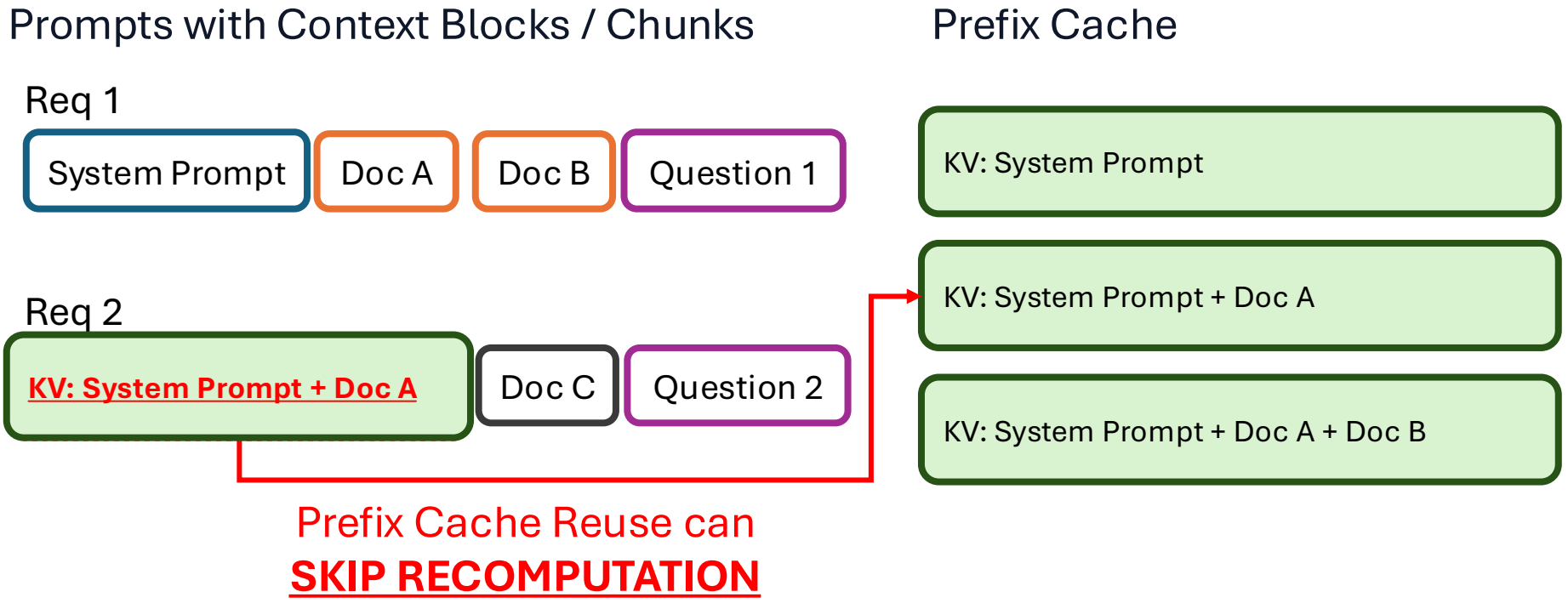
KV: System Prompt + Doc A

KV: System Prompt + Doc A + Doc B

Prefix Cache Can Improve Inference Efficiency



Prefix Cache Can Improve Inference Efficiency



Existing Approaches Trigger Cache Miss Or Harm Accuracy

Exact Prefix Matching:

RadixCache, LMCache, RAGCache, ...

Token Level Matching:

PromptCache, ...

Approximate KV Matching:

CacheBlend, Cache-Craft, ...

Existing Approaches Trigger Cache Miss Or Harm Accuracy

Exact Prefix Matching:

RadixCache, LMCache, RAGCache, …

Cached:

New:

Prefix cache miss

Token Level Matching:

PromptCache, …

Approximate KV Matching:

CacheBlend, Cache-Craft, …

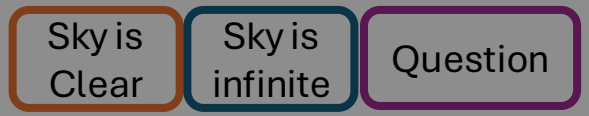
Existing Approaches Trigger Cache Miss Or Harm Accuracy

Exact Prefix Matching:

RadixCache, LMCache, RAGCache, ...

Cached:

New:

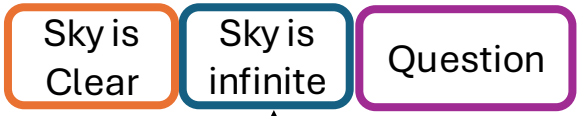


Prefix cache miss

Token Level Matching:

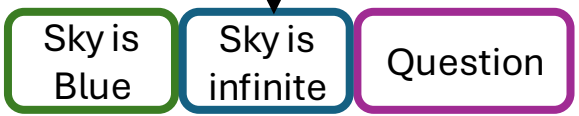
PromptCache, ...

Cached:



Reuse KV Directly

New:



Positional embedding mismatch

Accuracy degradation

Approximate KV Matching:

CacheBlend, Cache-Craft, ...

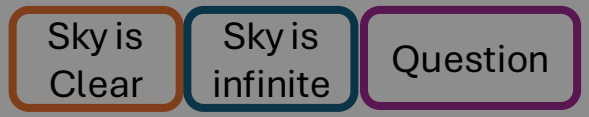
Existing Approaches Trigger Cache Miss Or Harm Accuracy

Exact Prefix Matching:

RadixCache, LMCache, RAGCache, ...

Cached:

New:

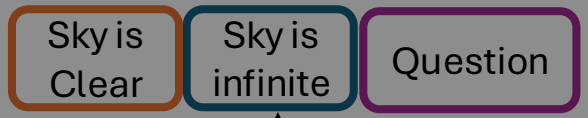


Prefix cache miss

Token Level Matching:

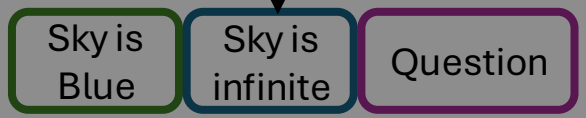
PromptCache, ...

Cached:



Reuse KV Directly

New:



Positional encoding mismatch

Accuracy degradation

Approximate KV Matching:

CacheBlend, Cache-Craft, ...

Cached:

Reuse base on threshold
Compute KV Distance

New:

Actual: Reuse Wrong KV → 'Blue' ✗

Accuracy degradation

Existing Approaches Trigger Cache Miss Or Harm Accuracy

Exact Prefix Matching:

RadixCache, LMCache, RAGCache, …

Cached:

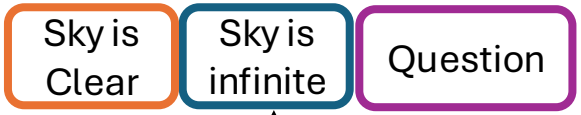
New:

Prefix cache miss

Token Level Matching:

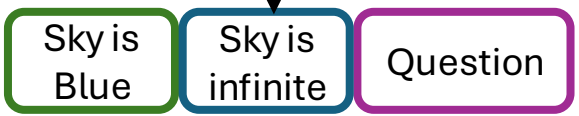
PromptCache, …

Cached:



Reuse KV Directly

New:



Positional encoding mismatch

Accuracy degradation

Approximate KV Matching:

CacheBlend, Cache-Craft, …

Cached:

Reuse based on threshold
Compute KV Distance

New:

Actual: Reuse Wrong KV → 'Blue' ✗

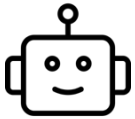
Accuracy degradation

We aim to achieve high reuse without sacrificing accuracy.

Key Observation 1:
Human Language Is Order-Robust

Request: “The brian can raed scrmableed wrods esaliy.”

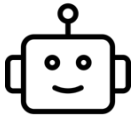
↓



Response: “The brain can read scrambled words easily.”

Request: “Genshin day I every play.”

↓



Response: “I play Genshin every day”

Typoglycemia *(a portmanteau of typo and hypoglycemia) The principle is that readers can comprehend text despite spelling errors and misplaced letters in the words.* —— *Wikipedia*

Scrambling *(linguistics): word order can vary without changing meaning, observed across many languages (Japanese, Latin, Russian, …) — Ross 1967.*

Key Observation 2:
Contexts Overlap in Multi-turn Conversation

Multi-turn conversations evolve gradually, with each turn building on the context of the previous. This is called **Topic shading [1].**

"When did Einstein win the Nobel Prize?"

Retrieve → Nobel Prize, Career milestones → “1921”

"What discovery earned him the Nobel?"

Retrieve → Scientific Discoveries, Nobel Prize → “Photoelectric effect”

40% doc overlap across turns (MT-RAG)

[1] *Jefferson, G. (1984). On stepwise transition from talk about a trouble to inappropriately next-positioned matters.*

Key Observation 3:
Contexts Overlap in Multi-session Scenario

Questions on the same topic tend to have overlapped contexts. This is called: **Lexical cohesion [1].**

" When did Einstein win the Nobel Prize?"

Retrieve → Nobel Prize, Career milestones

" What discovery earned Einstein the Nobel?"

Retrieve → Scientific Discoveries, Nobel Prize

Prefix cache hit

Context Overlap Rate:

79.2% of	**57.4% of**	**49.6% of**
MultihopRAG	NarrativeQA	QASPER

[1] Halliday, M. A. K., & Hasan, R. (1976). Cohesion in English. London: Longman.

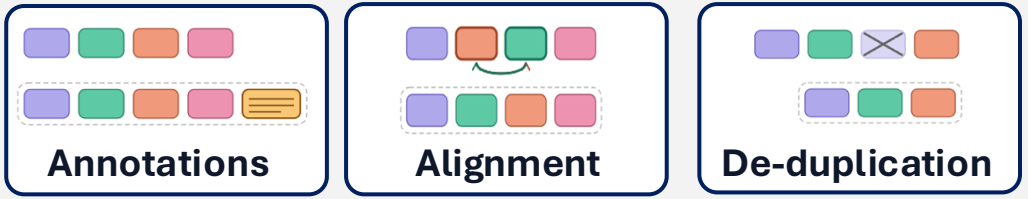
ContextPilot:
A Context Management System for Inference

Substantial reuse opportunities emerge only at the context level, not in KV cache – **a new design space!**

Context Reuse:

1. **Cross-Request Reuse**
   align context to hit prefix cache across requests
2. **Cross-Turn Reuse**
   carry context across conversation turns, remove duplicates

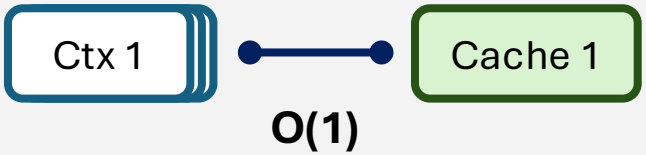
Three Key Mechanisms on Context



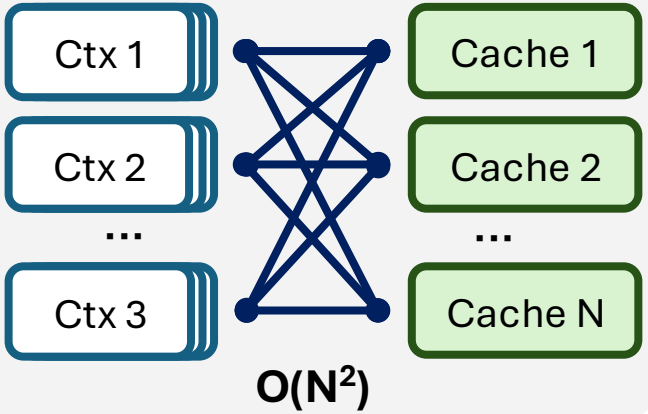
ContextPilot — a **context management system** for inference engines

The Challenge of Finding Reusable Contexts

1 stored context
the best match is trivial



N stored contexts
must scan all to find best match



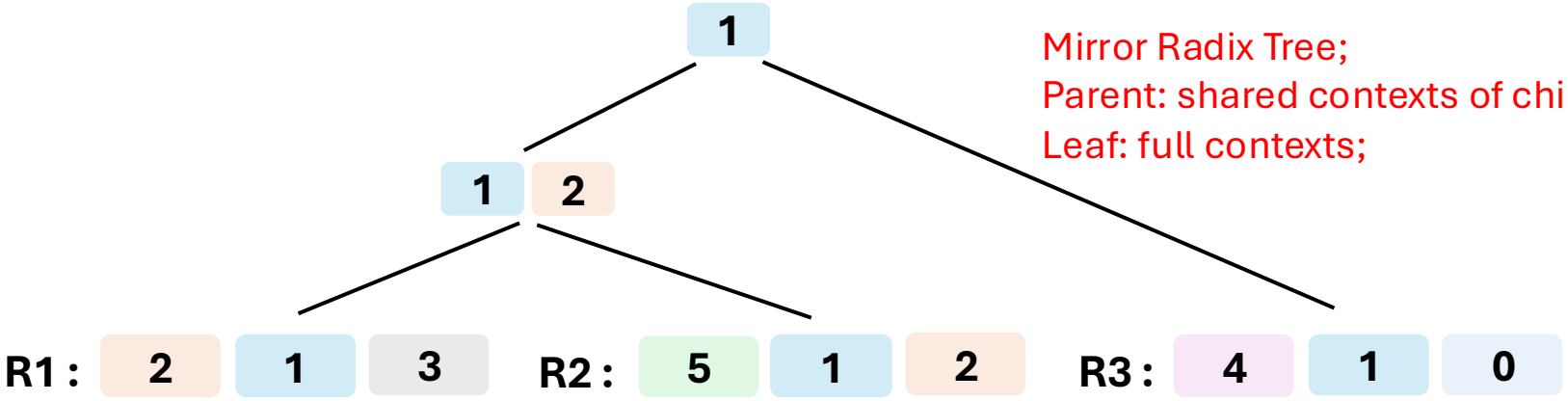
An index is needed to efficiently
find the best-matching context to reuse

Contribution 1: Context Block (CB) Index

Request 1 :	CB **2**, CB **1**, CB **3**
Request 2 :	CB **5**, CB **1**, CB **2**
Request 3 :	CB **4**, CB **1**, CB **0**

R1 :	**2**	**1**	**3**
R2 :	**5**	**1**	**2**
R3 :	**4**	**1**	**0**

Contribution 1: Context Index



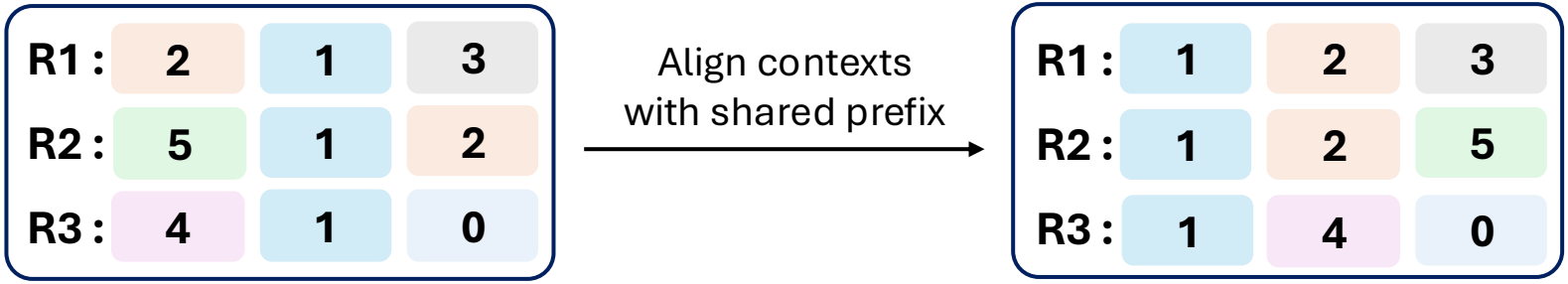
Mirror Radix Tree;
Parent: shared contexts of children;
Leaf: full contexts;

Distance Function:

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

Contribution 2: Context Alignment & Annotation

Context Alignment:

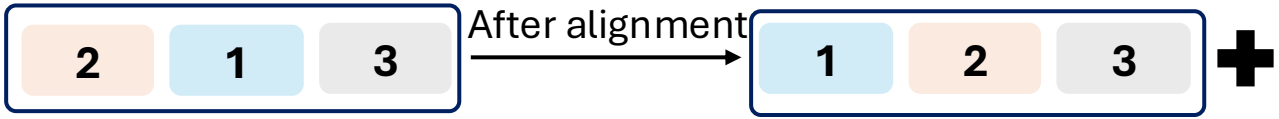


Before: **5%** Cache Hit (System Prompt)

After: **33%-67%** Cache Hit

Accuracy loss from alignment is about **0.1–3.3%.**

Order Annotation:



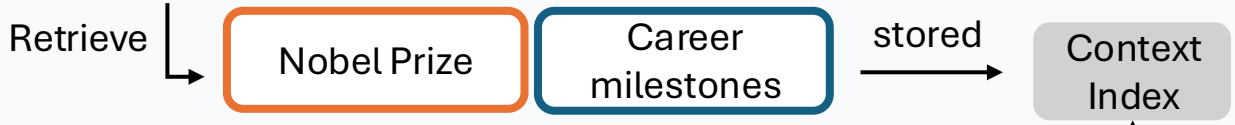
"Please read in priority order: [2] > [1] > [3]"

Negligible token overhead, recovers lost accuracy

Contribution 3: Context De-duplication & Annotation

Context De-duplication:

Turn 1: "When did Einstein win the Nobel Prize?"



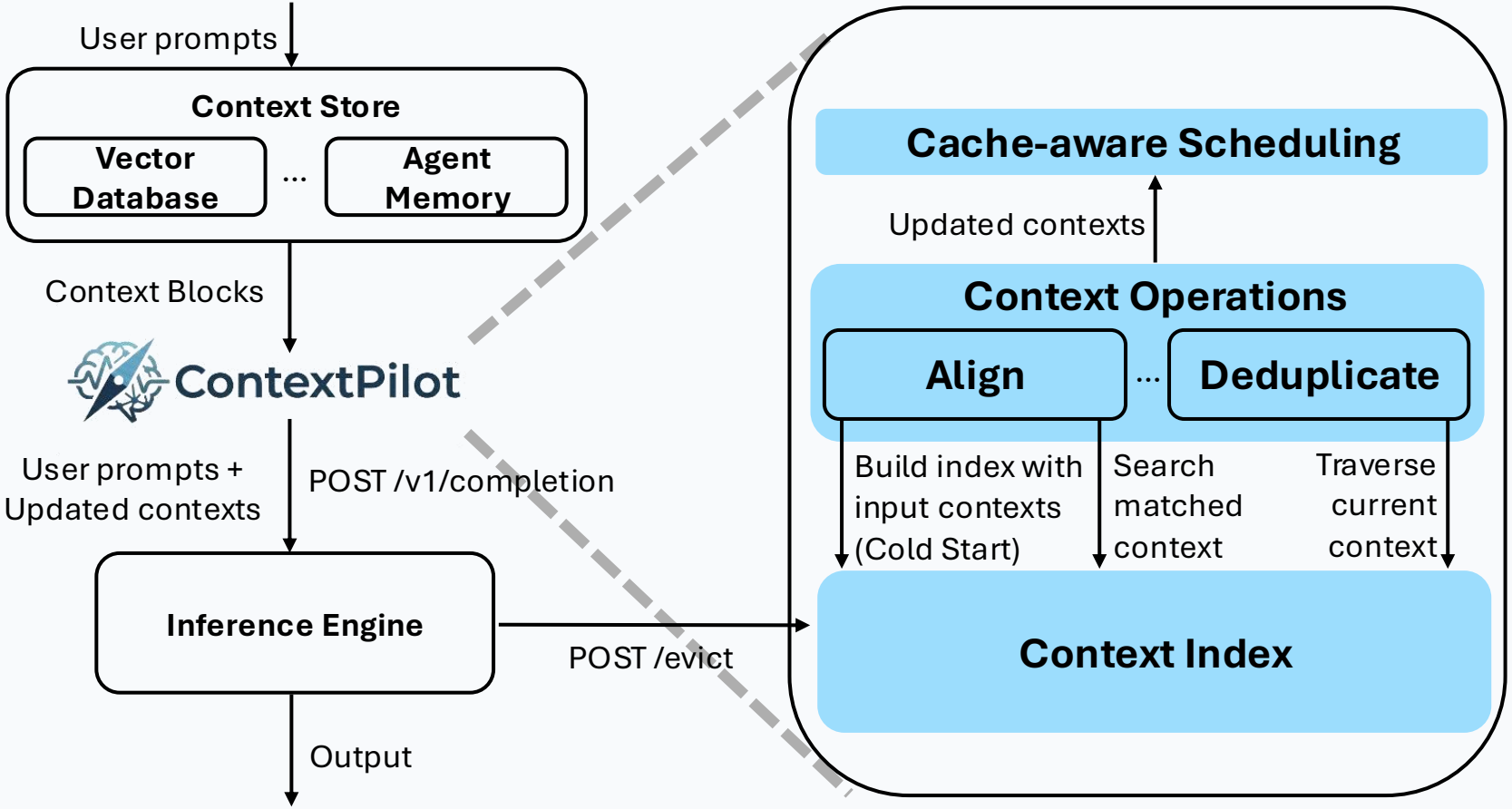
Turn 2: "What discovery earned him the Nobel?"

Accuracy loss from deduplication is about 0.4-0.9%.

Location Annotation:

Turn 2: "What discovery earned him the Nobel?"

ContextPilot: System Architecture



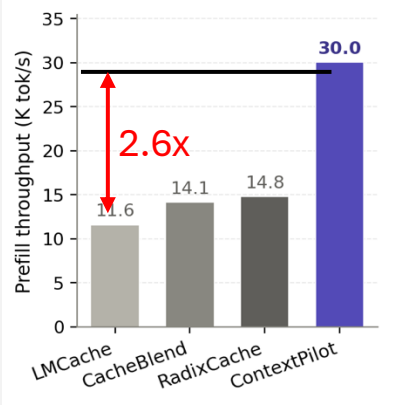
Results: Prefill Speedup Across Diverse Workloads

RAG workload:

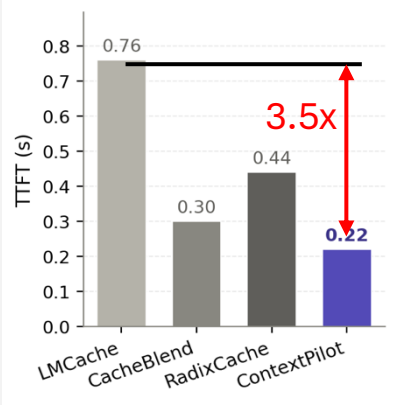
- Offline large batch -- **2.6x** Prefill throughput speedup
- Online multi-turn -- **3.5x** TTFT reduction

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

Offline large batch (DeepSeek-R1)



Online multi-turn (Qwen3-4B)



Agentic Workload (OpenClaw):

- ContextPilot achieves **1.8-3.8x** TTFT speedup

Metrics	Method	Doc 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 at Scale

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 recover accuracy loss of alignment

Scaling to large MoE models such as DeepSeek-R1 (671B)

ContextPilot accelerates the latest SGLang prefill by **1.8x** on MultihopRAG for DeepSeek-R1 (16 GPUs), with **full accuracy preserved**.

ContextPilot Integrates Across the LLM Stack

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

Takeaways

ContextPilot explores a new design space for context optimization

- Solid design assumption rooted to linguistic and empirical studies
- Compatible system designs for both cluster and edge deployments

What's next:

- Knowledge-aware cache reuse;
- Multi-modal use cases;
- Broader agent framework support: NanoClaw, Codex, …

Github

Contact:
ysc.jiang@ed.ac.uk

github.com/EfficientContext/ContextPilot

openclaw plugins install @contextpilot-ai/contextpilot

hermes plugins install EfficientContext/ContextPilot

ContextPilot