KV Cache Reuse for Elastic LLM Inference on Edge Devices

Peishuo Wang, Zhenzhe Zheng, Xiaoyao Huang[†], Jie Wu[†*], Fan Wu, Guihai Chen
Shanghai Jiao Tong University & [†]China Telecom & [*]Temple University

2026-6-22

IEEE ICDCS 2026, Seoul, South Korea

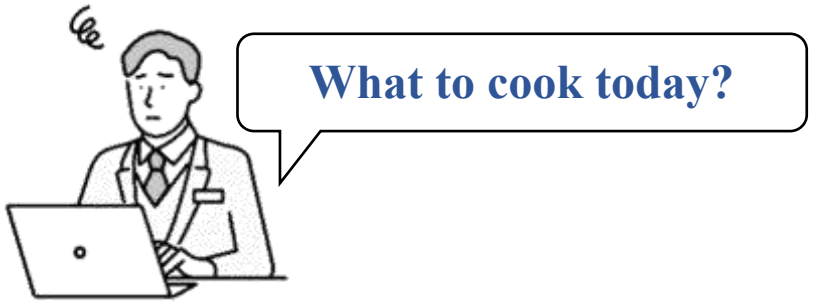
Outline

1. Background & Motivation
2. Design of ElastKV
3. Improvements for Practical Deployment
4. Evaluation
5. Conclusion

Background & Motivation

Background: On-Device LLM Services

LLMs deployed on edge devices for privacy, low latency, and personalization



personalized data

Stock in Smart Fridge
bell pepper, brocoli, carrot

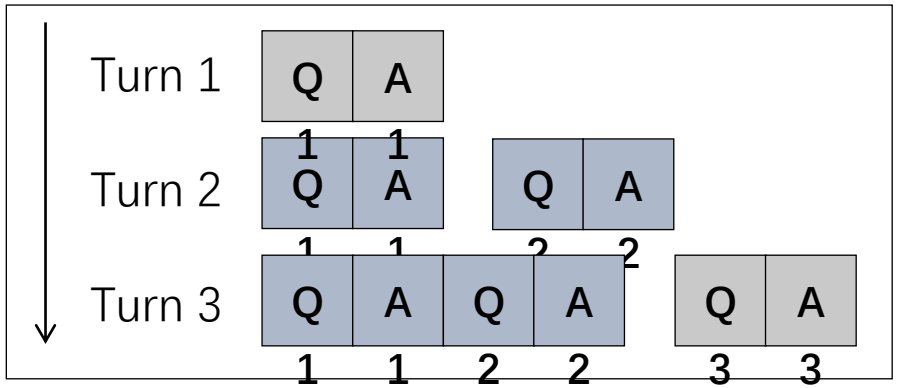
History Orders
2026-6-20, *Mexican Cuisine*

Chat Records
… I'm allergic to peanuts …

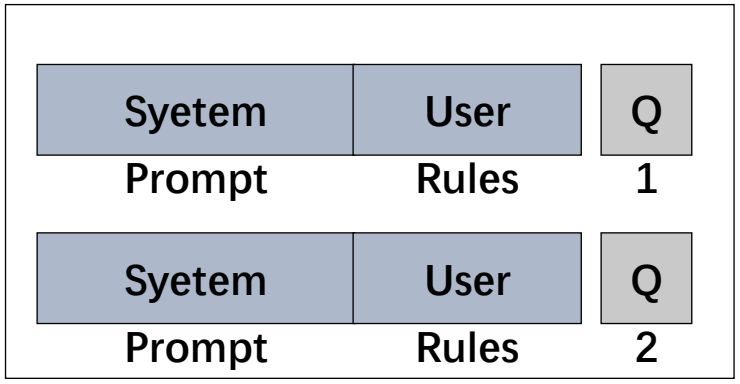
general knowledge

Prefix Caching: reuses KV pairs from common prefixes across requests

Multi-trun QA



Shared prefix



Reused KV

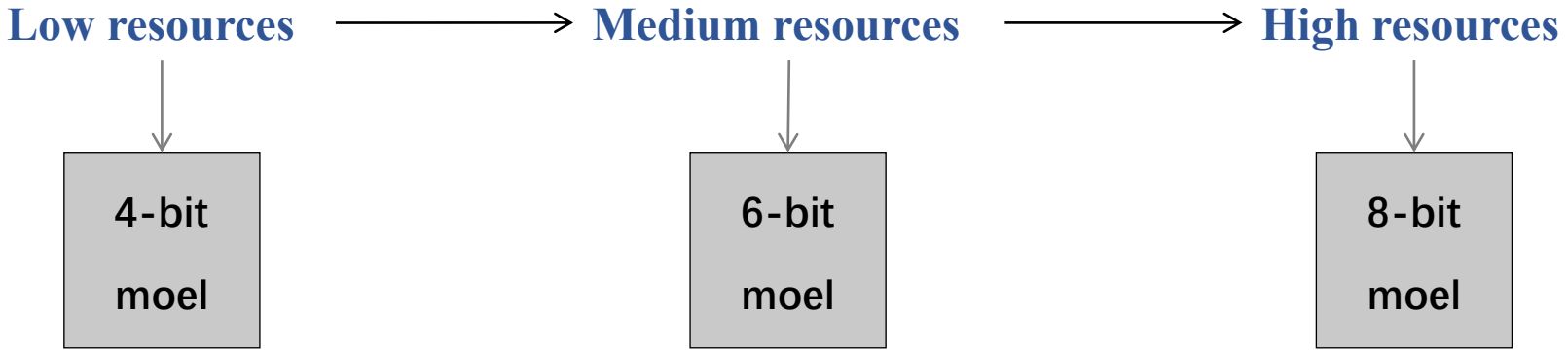
Prefilling

Background: Elastic LLM Inference on Edge Devices

Fluctuating resources: edge devices often experience fluctuating memory and compute resources

Fluctuating resources makes elastic inference essential for on-device LLM deployment.

Elastic Inference: dynamically switches between different versions of the model



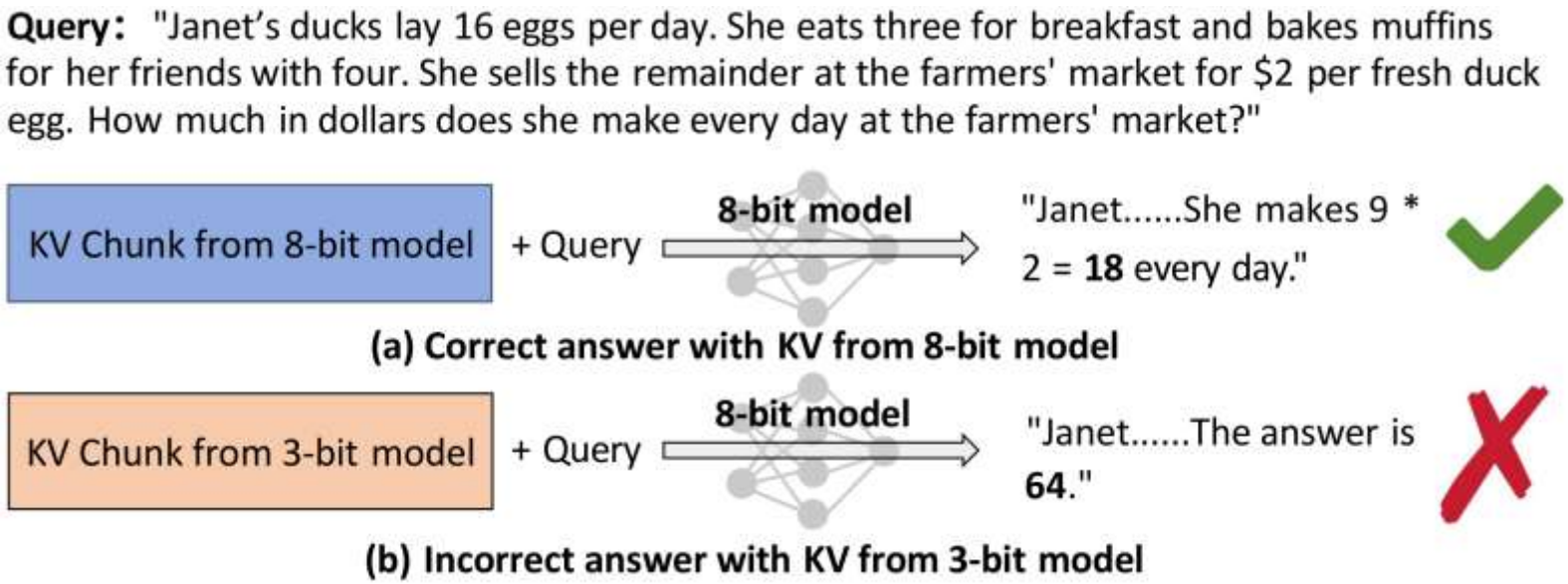
Challenge: How to support both KV cache reuse in elastic LLM inference?

Motivation: The Cross-Precision KV Reuse Problem

When resources fluctuate, system switches model precision:
High memory → 8-bit model (higher accuracy)
Low memory → 3-bit model (lower footprint)

Prior KV caches may come from different precision models.

Directly reusing KV from lower-precision → wrong answer!



Key Observations

Observation 1:

Low-precision KV → harms high-precision model

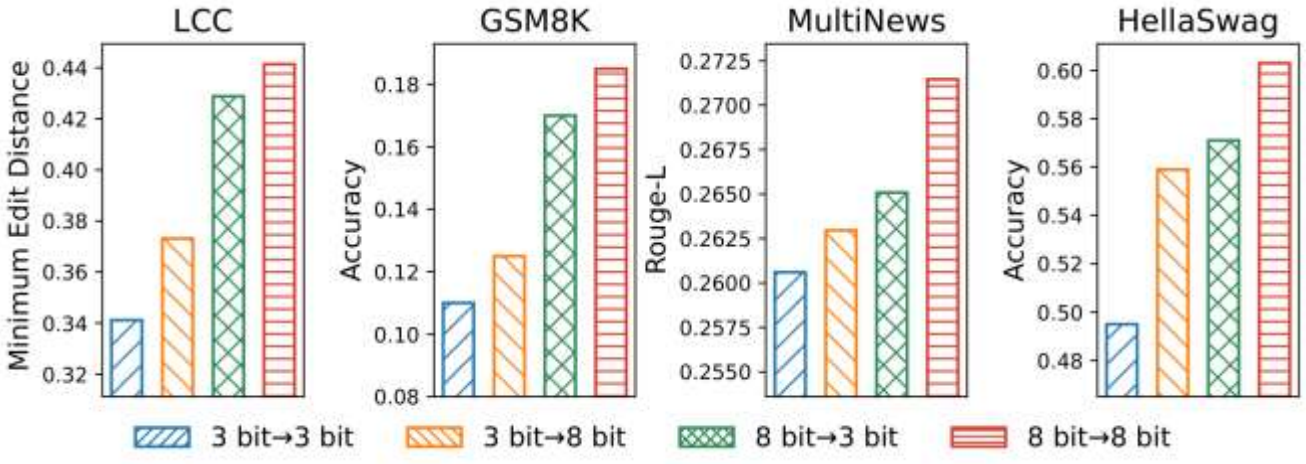
High-precision KV → can improve low-precision model

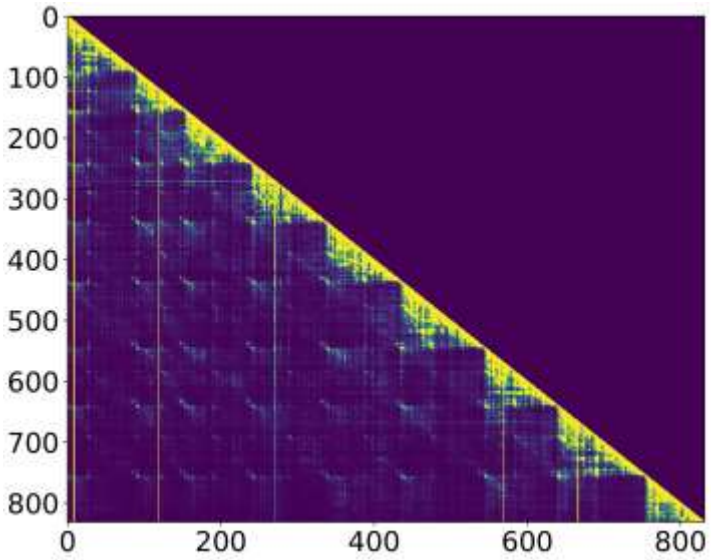
Observation 2:

Attention sparsity: few tokens receive high attention

LLMs can tolerate small KV deviations

→ Selective recomputation can correct most deviations!





Challenges

Challenge 1: Cross-Precision Caching

- The reliability of reusing or recomputing them is unclear. A recomputation rule is needed.

Challenge 2: Recomputation Across Models

- Existing KV recomputation methods are designed for single model scenarios.
- Quantifying each token's KV deviation impact on output — output tokens unavailable.

Challenge 3: Overhead Control

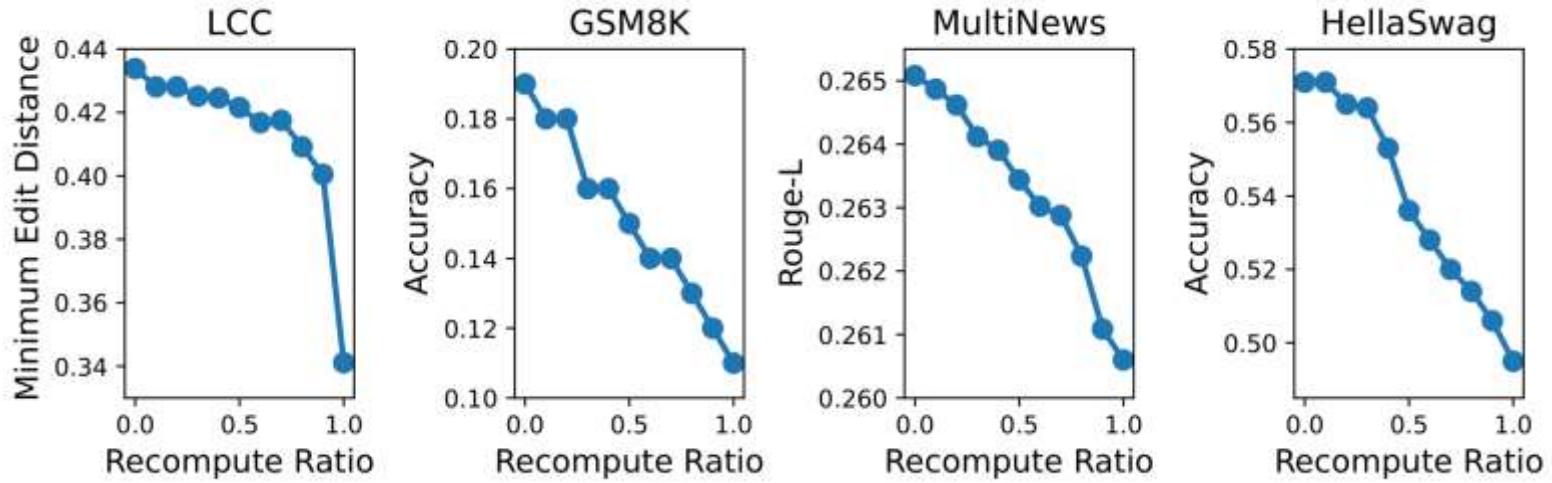
- Accurate estimation requires two inference rounds — impractical for edge devices.
- Careful control of extra computational overhead, especially on edge devices.

Design of ElastKV

Design: Insights and Recomputation Rule

Impact of Recomputing KV from Higher-bit Models:

Recomputing KV from higher-precision with lower-precision model degrades performance.



Rule: $R(Bt, bt)_i = 1$ if $bt > Bt,i$

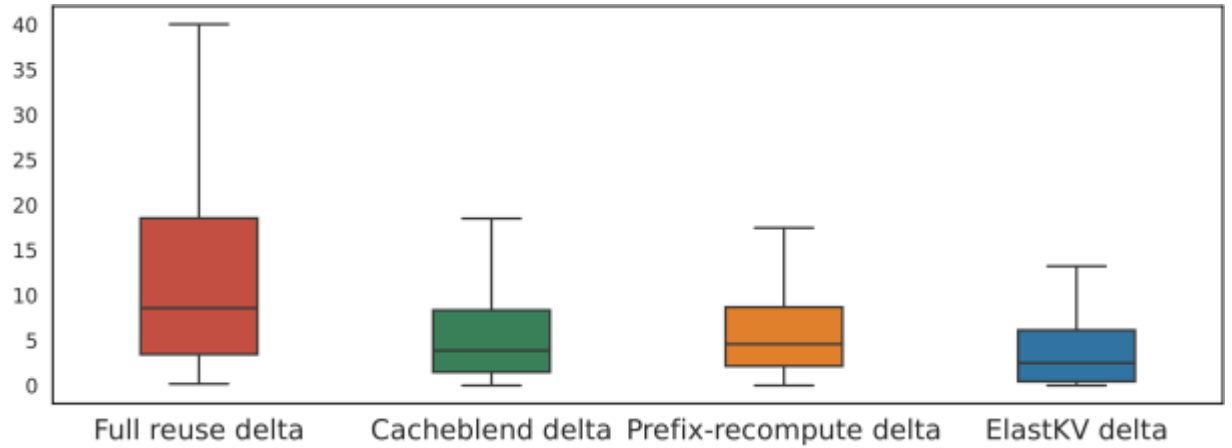
(only recompute when current model is higher precision)

- Fully reuse KV from higher-precision models
- Selectively recompute KV from lower-precision models

Design: Token Priority Estimation - Taylor approximation

Drawbacks of Prior Work:

KV delta degrades model performance. Existing methods fail to sufficiently reduce it as they only considered repairing position encoding error and cross attention error.



Estimate the impact of each token's KV deviation on subsequent tokens:

- We use first-order Taylor expansion:

$$\mathcal{I}_{ij} \approx \alpha_{ij}\Delta v_j + \alpha_{ij} \cdot \frac{q_i}{\sqrt{d}} \cdot (v_j - y_i) \cdot (\Delta k_j).$$

(I_ij: influence of token j's KV deviation on output at position i)

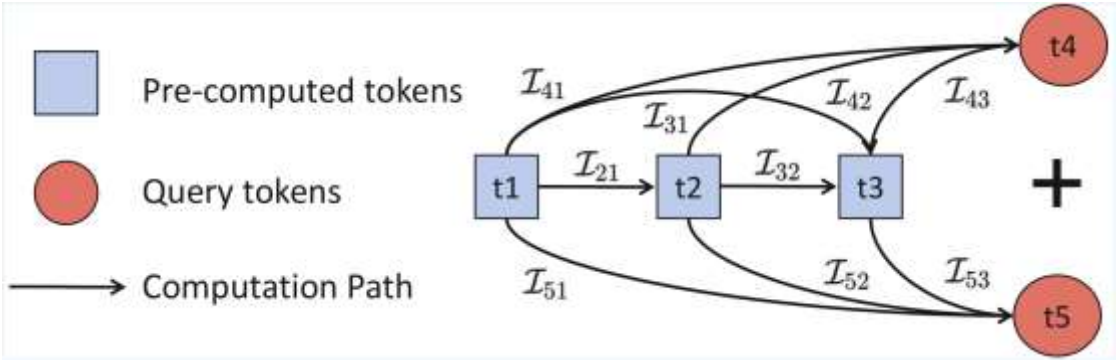
Then we compute the **L2 norm** to get scalar values for easier comparison.

- Avoids the need for two inference rounds

Design: Token Priority Estimation - Influence Path Propagation

Estimate impact through all paths:

- KV delta has **cascading influence** on subsequent tokens



We combine all influence paths to get more precise impact on subsequent tokens.

- Estimating the impact of each prior token **on the final output**

  We estimate it by estimating their influence on each token within the query(user input).

$$\mathcal{A}'_i = \sum_{m=s+q_t-min(q_t,L_w)+1}^{s+q_t} \mathcal{I}_{mi} + \sum_{n=i+1}^{s} \mathcal{I}_{ni}\mathcal{A}'_n,$$

$$\mathcal{A}_i = \mathcal{A}'_i \cdot D_i.$$

Design: Selective KV Recomputation Workflow

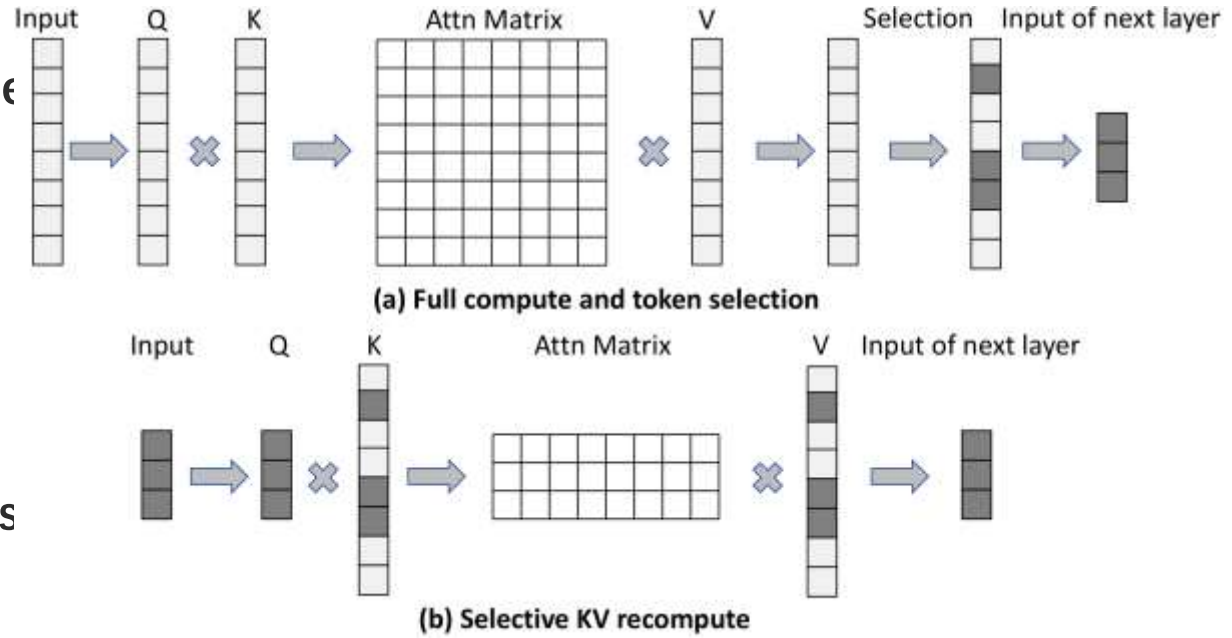
Step 1: Apply recomputation rule → candidate se Ct

Step 2: Estimate priority At at shallow layer

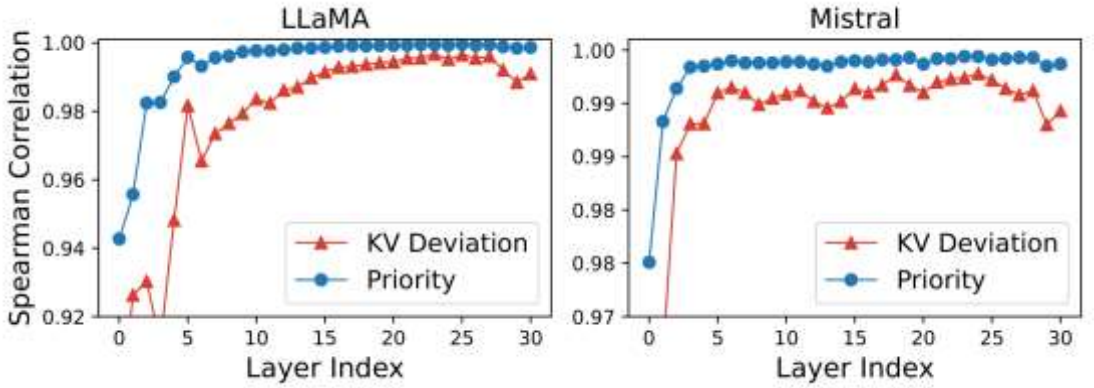
(Taylor + influence paths)

Step 3: Select top r% tokens from Ct by priority

Step 4: Recompute KV for selected, reuse the res



Token selection fixed across layers (Spearman > 0.92)



Improvements for Practical Deployment

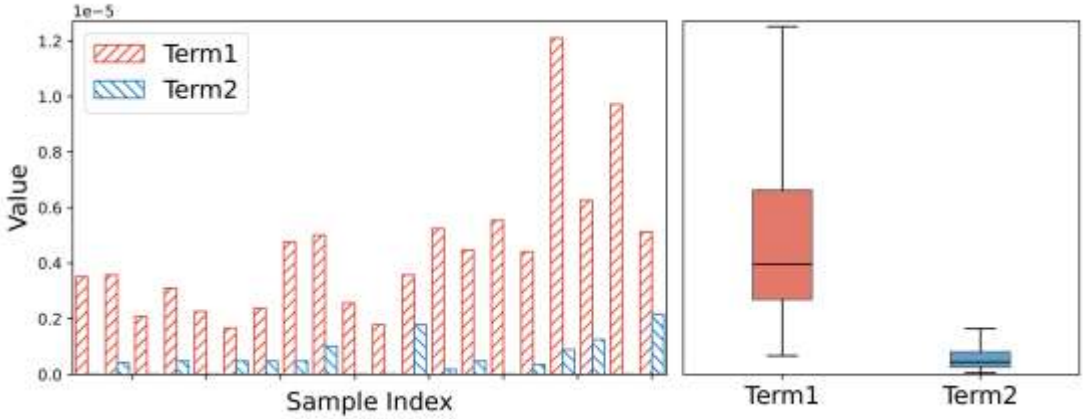
Taylor Simplification:

- First term dominates → Retain term1 only

$$\mathcal{I}_{ij} \approx \alpha_{ij}\Delta v_j + \boxed{\alpha_{ij} \cdot \frac{q_i}{\sqrt{d}} \cdot (v_j - y_i) \cdot (\Delta k_j).}$$

Omit

$$\mathcal{I}_{ij} \approx \alpha_{ij}\Delta v_j$$



Memory-Efficient L2 Norm:

- precompute L2 norm to avoid 256GB tensor (for 7B model)

$$\mathcal{I} := \sqrt{\sum_{dim}(\alpha \cdot \Delta v)^2} = \alpha \cdot \sqrt{\sum_{dim}(\Delta v)^2} = \alpha \cdot \|\Delta v\|_2 .$$

- If use both terms, we omit cross-product term

  Reason : near-orthogonality (cos ≈ -0.009)

$$\mathcal{I} := \sqrt{\|term1\|_2^2 + \|term2\|_2^2 \boxed{+ 2\langle term1 \cdot term2 \rangle}.}$$

Omit

Experiments

Experiment Setup

Models:

- LLaMA2-7B
- Mistral-7B
- TinyLLaMA-1.1B
- OPT-350M

Hardware:

Server:

- RTX 3090 (24GB)

Edge devices:

- Laptop: RTX 4060 (8GB)
- Laptop: GTX 1650 (4GB)

Datasets:

- HellaSwag (commonsense reasoning)
- GSM8K (mathematical reasoning)
- LCC (code generation, ~4K tokens)
- MultiNews (long context summarization, ~4K tokens)

Baselines:

Full KV Recomputation

Full KV Reuse

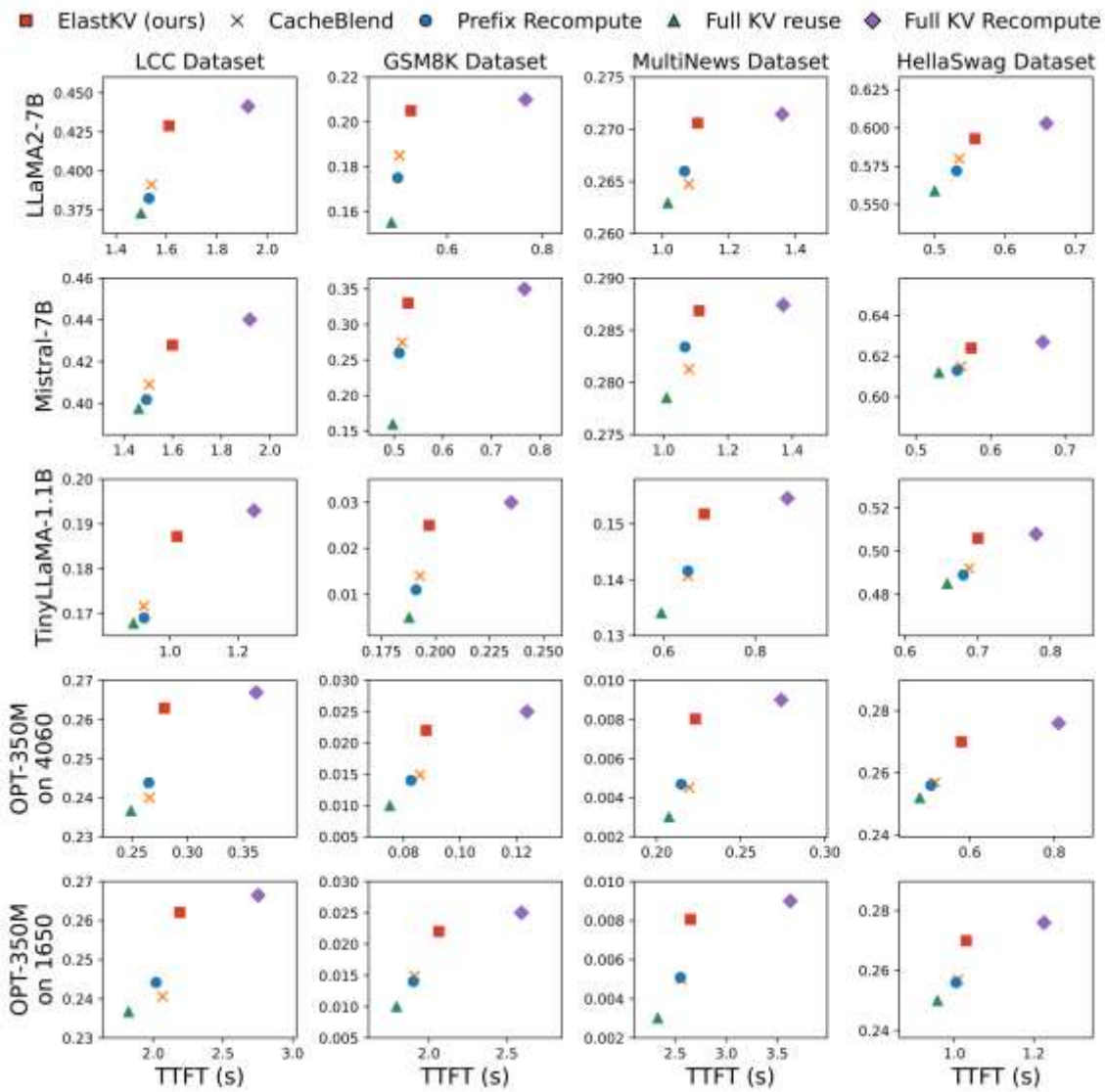
CacheBlend (Selective Recomputation)

EPIC (Prefix Recomputation)

Quantization method:

K-means quantization (from Any-Precision-LLM)

Main Results: Speed-Quality Trade-off

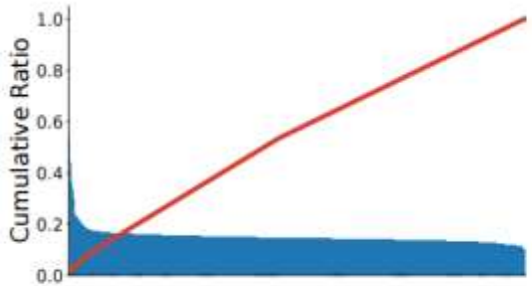


Only 3-bit -> 8 bit

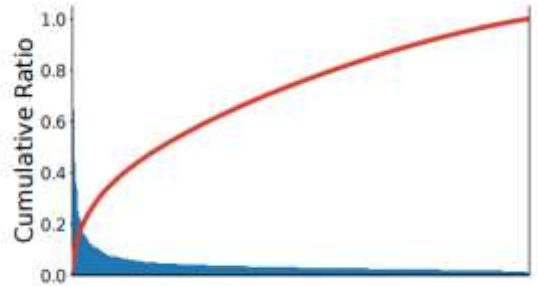
Key Results:

≤2% accuracy loss

~3× speedup vs. full recomputation



(a) KV Deviation



(b) ElastKV Priority

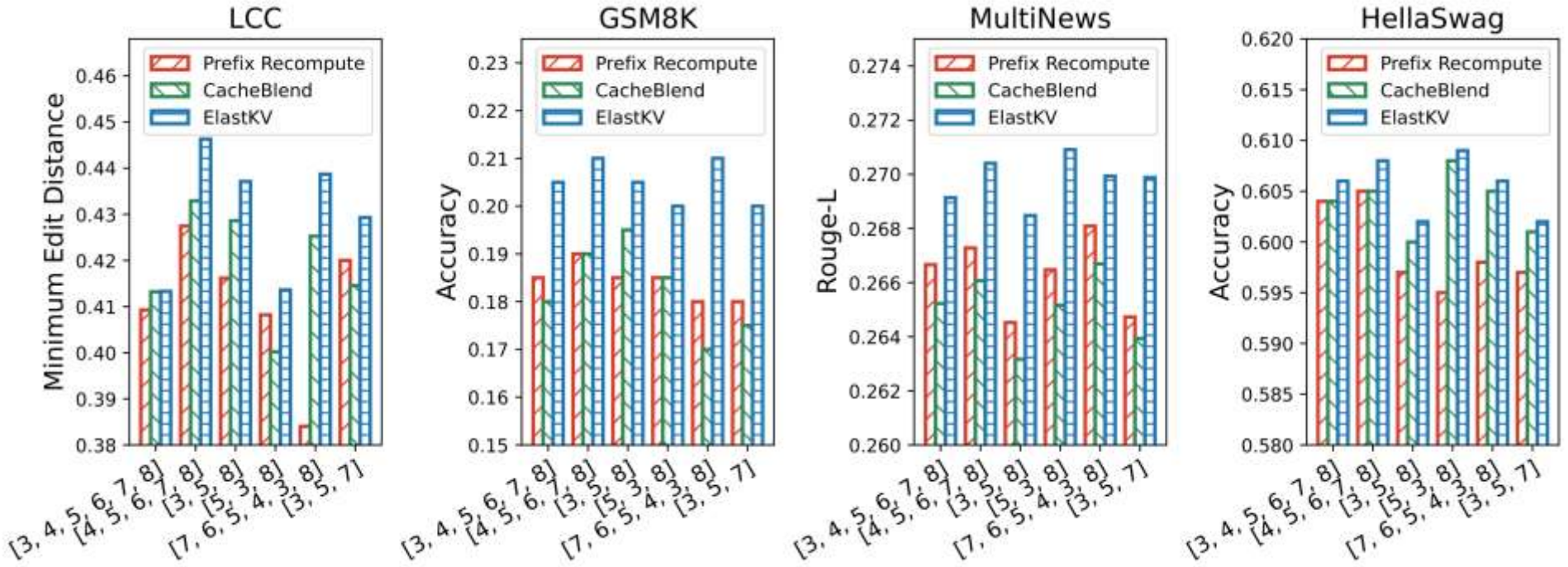
Sensitivity Analysis: Multi-Precision & Varying Ratios

Varying Chunk Numbers and Precisions.

This setup simulates elastic inference under resource fluctuations.

Context is partitioned into more chunks, each processed by a model of distinct precision.

Only LLaMA2-7B

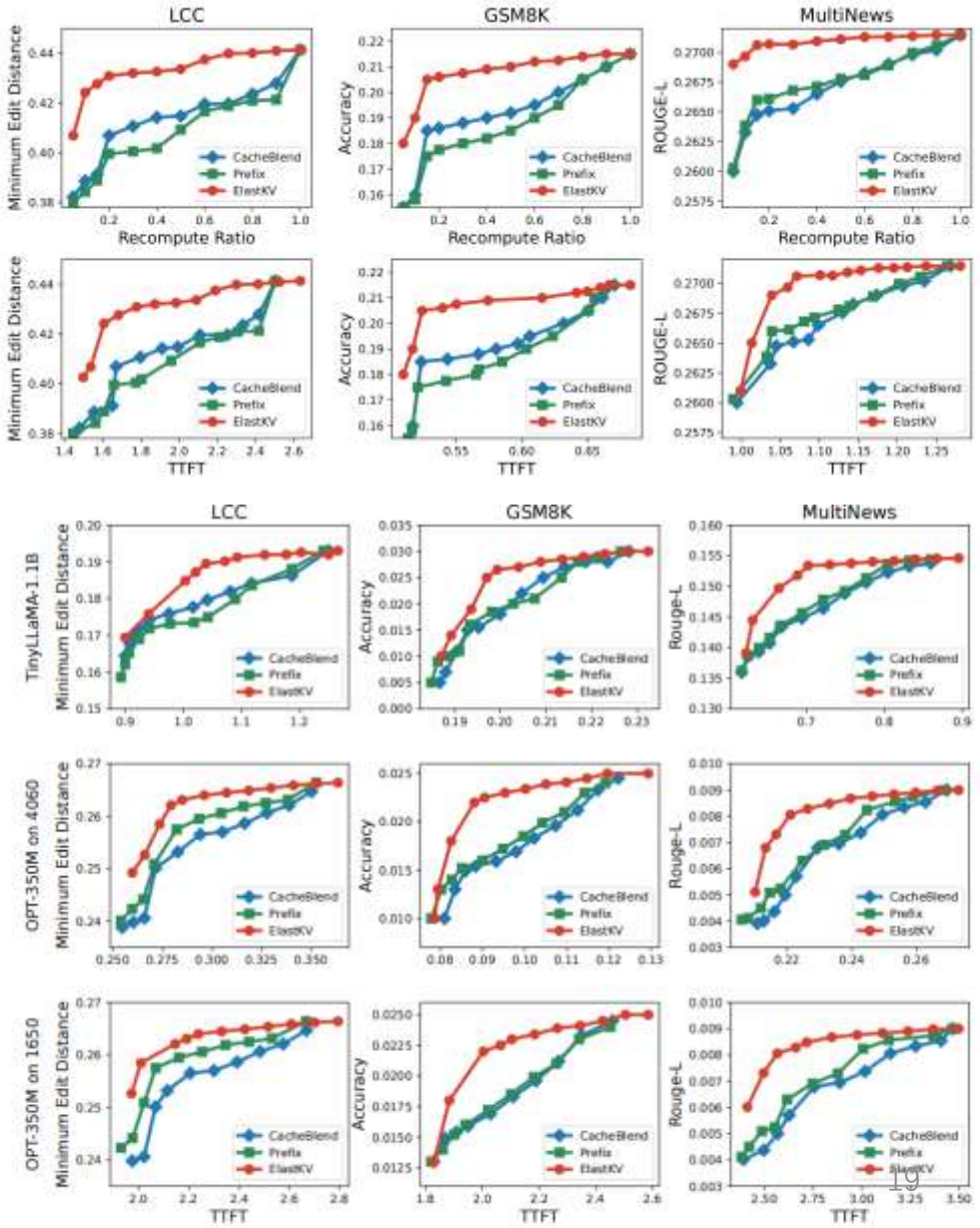


ElastKV adapts better to precision switching than previous approaches

Sensitivity Analysis: Multi-Precision & Varying Ratios

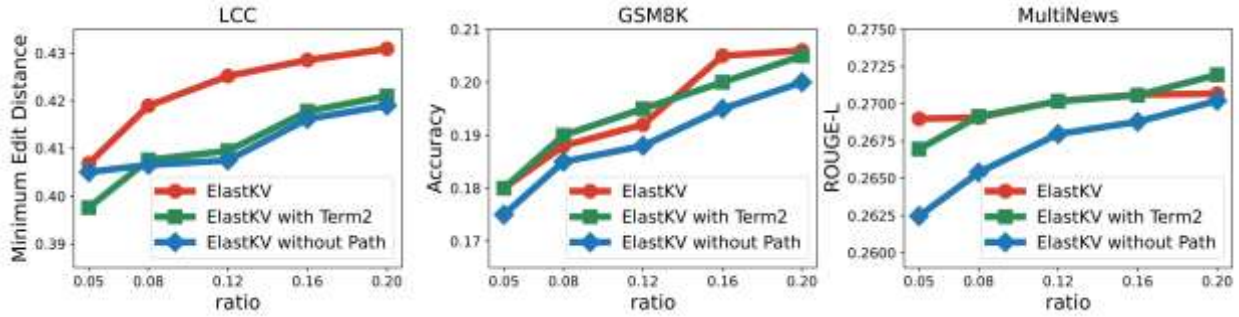
Varying Recompute Ratios.

- We compare ElastKV and baselines across varying recompute ratios and TTFT (time to first token) on three datasets, evaluated with LLaMA2-7B.
- Since inputs in HellaSwag are very short, we do not include HellaSwag in this experiment
- At a very low ratio, ElastKV exhibits a worse trade-off due to extra estimation overhead. However, such a low ratio is not practical.
- With practical ratios, ElastKV outperforms previous approaches



Ablation Study

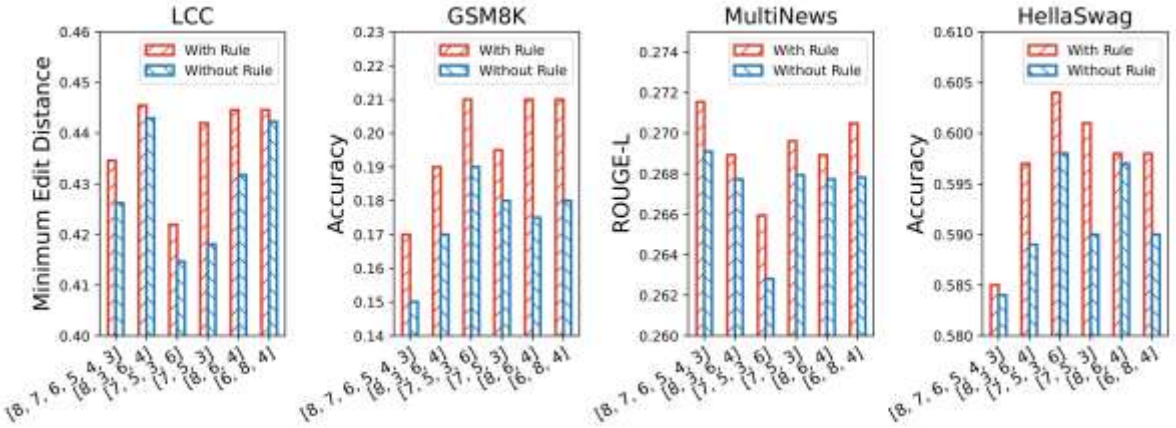
Using the second term of the Taylor expansion & Without Considering Influence Paths.



Influence path: removes → performance drops

Disabling the Recomputation Rule.

Term2: Very few improvements



Without rule: degrades on high-bit KV

With rule: prevents unnecessary degradation

Conclusion

Conclusion

ElastKV: first work to address KV cache reuse from mixed-precision models for elastic LLM inference on edge devices

Recomputation Rule: only recompute when current model precision > cached KV precision

Token Priority Estimation: Taylor approximation + influence-path propagation

Results: ⩽2% accuracy loss + ~3× speedup; better trade-off than all SOTA methods

ElastKV enables practical on-device LLM serving under dynamic resource constraints.

Thanks for Your Attention!

KV Cache Reuse for Elastic LLM Inference on Edge Devices

Peishuo Wang, Zhenzhe Zheng, Xiaoyao Huang, Jie Wu, Fan Wu, Guihai Chen

Shanghai Jiao Tong University & China Telecom & Temple University