EdgePrompt: A Distributed Key-Value Inference Framework for LLMs in 6G Networks

Jiahong Ning, Pengyan Zhu, Ce Zheng, Gary Lee, Sumei Sun, Tingting Yang

IEEE INFOCOM 2025 PerAI-6G Workshop

Dalian Maritime University

Motivation

The Challenges of LLM Inference in 6G

- LLMs offer new capabilities in 6G (e.g., chatbots, intelligent routing)
- But: High inference latency
- And: Privacy risks from cloud-based processing
- Existing solutions fall short
- ➤ Need new inference architecture balancing speed and privacy

Our Contribution

- EdgePrompt: A cloud-edge collaborative LLM inference architecture
- Separation of cloud and edge prompts to enhance privacy and efficiency
- KV pair synchronization for distributed attention fusion
- Theoretical model optimizing latency via overlap of communication and computation
- Extensive experiments: Higher throughput, lower latency, better scalability

Inference Bottlenecks

Inference

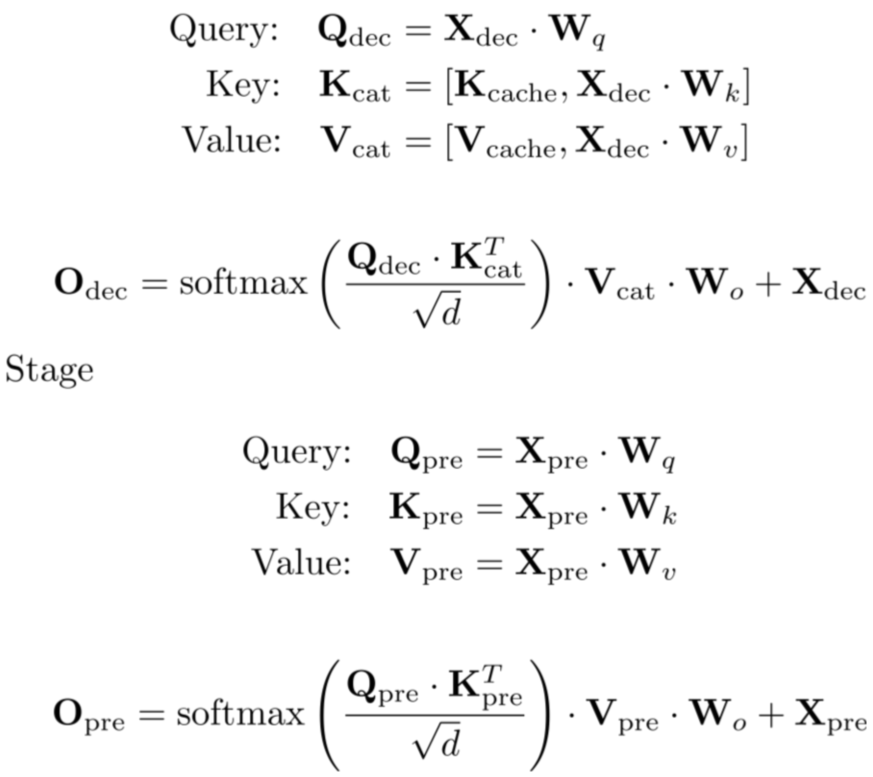
- Prompting stage: parallel computation of Q/K/V
- Generation stage: sequential token-by-token decoding

KV cache growth → increases memory & bandwidth usage

Bottlenecks in:

- Long input sequences
- High concurrency scenarios

Existing methods (PagedAttention, Prompt Cache) reduce memory usage, but not latency or privacy risks.



$$\begin{aligned} \text{Query:} \quad \mathbf{Q}_{\text{dec}} &= \mathbf{X}_{\text{dec}} \cdot \mathbf{W}_q \\ \text{Key:} \quad \mathbf{K}_{\text{cat}} &= [\mathbf{K}_{\text{cache}}, \mathbf{X}_{\text{dec}} \cdot \mathbf{W}_k] \\ \text{Value:} \quad \mathbf{V}_{\text{cat}} &= [\mathbf{V}_{\text{cache}}, \mathbf{X}_{\text{dec}} \cdot \mathbf{W}_v] \end{aligned}$$

$$\mathbf{O}_{\text{dec}} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{dec}} \cdot \mathbf{K}_{\text{cat}}^T}{\sqrt{d}}\right) \cdot \mathbf{V}_{\text{cat}} \cdot \mathbf{W}_o + \mathbf{X}_{\text{dec}}$$

Stage

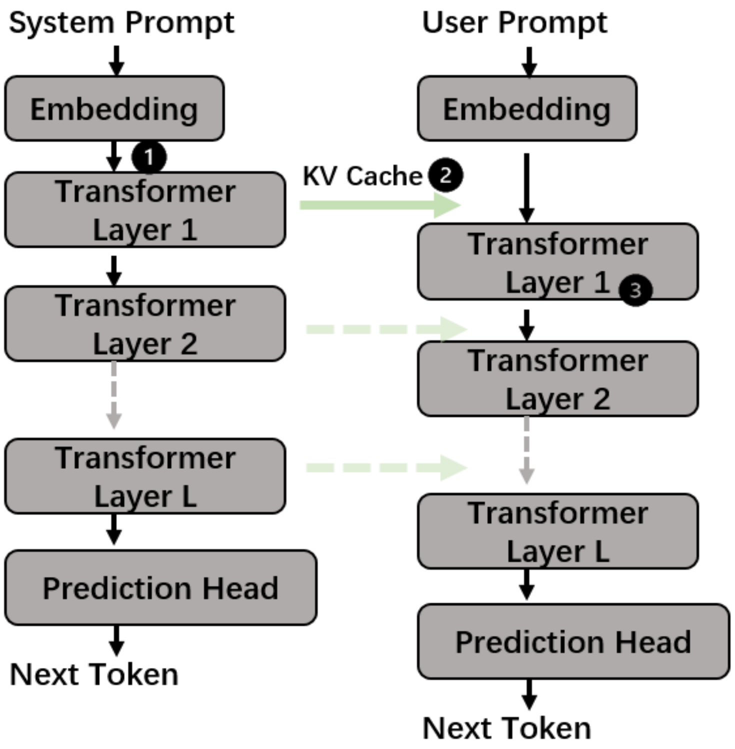
$$\begin{aligned} \text{Query:} \quad \mathbf{Q}_{\text{pre}} &= \mathbf{X}_{\text{pre}} \cdot \mathbf{W}_q \\ \text{Key:} \quad \mathbf{K}_{\text{pre}} &= \mathbf{X}_{\text{pre}} \cdot \mathbf{W}_k \\ \text{Value:} \quad \mathbf{V}_{\text{pre}} &= \mathbf{X}_{\text{pre}} \cdot \mathbf{W}_v \end{aligned}$$

$$\mathbf{O}_{\text{pre}} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{pre}} \cdot \mathbf{K}_{\text{pre}}^T}{\sqrt{d}}\right) \cdot \mathbf{V}_{\text{pre}} \cdot \mathbf{W}_o + \mathbf{X}_{\text{pre}}$$

EdgePrompt: A Distributed Inference Architecture

- Split input prompt into:
  - Cloud Prompt: general instructions, processed in cloud
  - Edge Prompt: user data, processed locally

- Sensitive information remains on-device
- Cloud handles heavy computation, edge ensures privacy
- KV Cache transferred from cloud to edge for downstream layers
- Optimizes both efficiency and privacy



KV-Based Attention Fusion

- Attention computed separately for cloud and edge prompts:

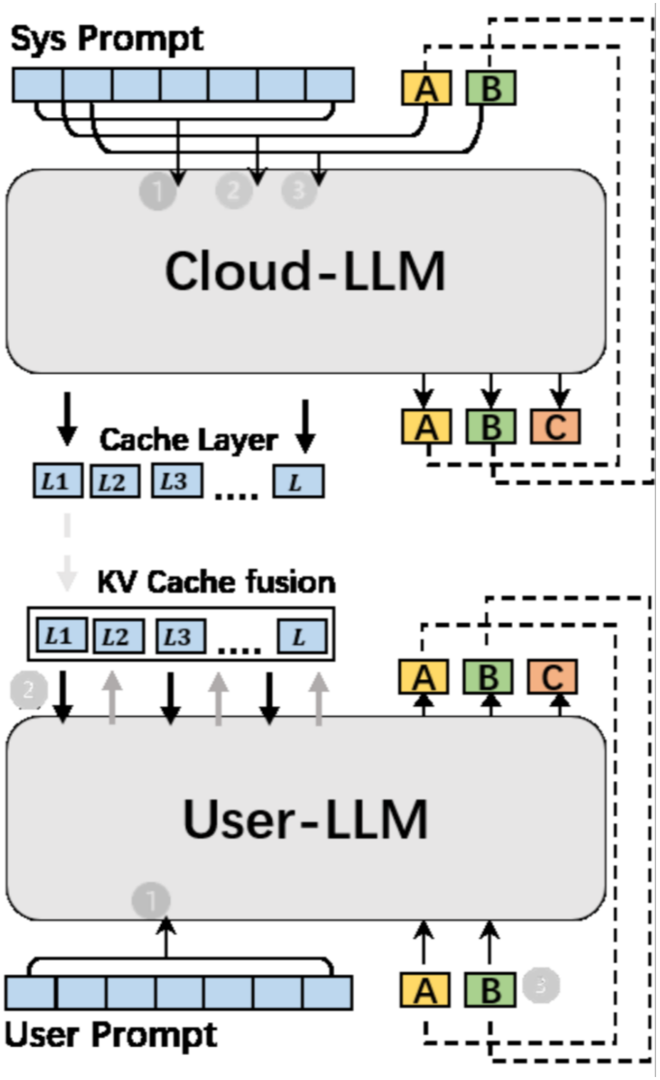
$$A^l_{\text{cloud}} = \text{softmax}\left(\frac{Q^l_{\text{cloud}} K^{lT}_{\text{cloud}}}{\sqrt{d}}\right) V^l_{\text{cloud}}$$

$$A^l_{\text{edge}} = \text{softmax}\left(\frac{Q^l_{\text{edge}} K^{lT}_{\text{edge}}}{\sqrt{d}}\right) V^l_{\text{edge}}$$

- Fused at each decoder layer using:

$$O^l_{\text{module}} = \alpha_{\text{cloud}} A^l_{\text{cloud}} + \alpha_{\text{edge}} A^l_{\text{edge}}$$

- α balances cloud vs. edge influence
- Efficient reuse of cloud KV → avoids redundant computation

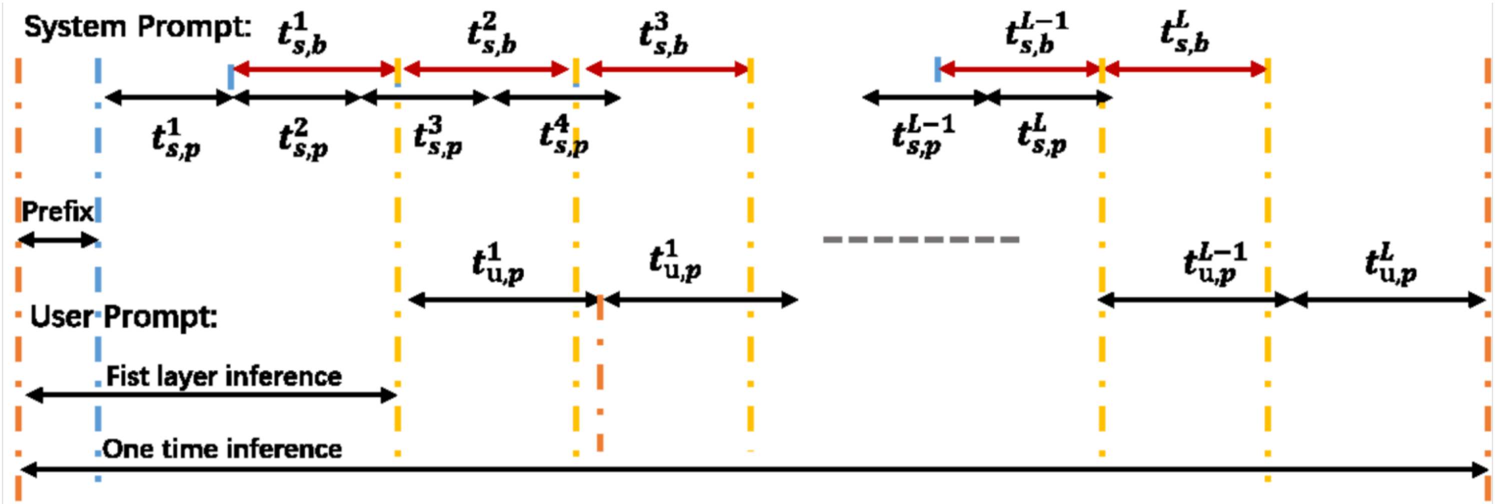


Communication Model – Overlapping for Latency Reduction

- Latency-Aware Communication Model
  - Key challenge: Cloud-edge coordination may introduce delay
  - Solution: Overlapping computation and communication
  - Total latency:

$$T = T_{\text{prefix}} + (t_{1c,c} + t_{1c,t}) + \max\left[\sum_{l=2}^{L}(t_{lc,c} + t_{lc,t}), \sum_{l=1}^{L} t_{le,c}\right]$$

- Optimize based on three cases:
  - P1: Transmission-bound
  - P2: Edge-compute-bound
  - P3: Mixed



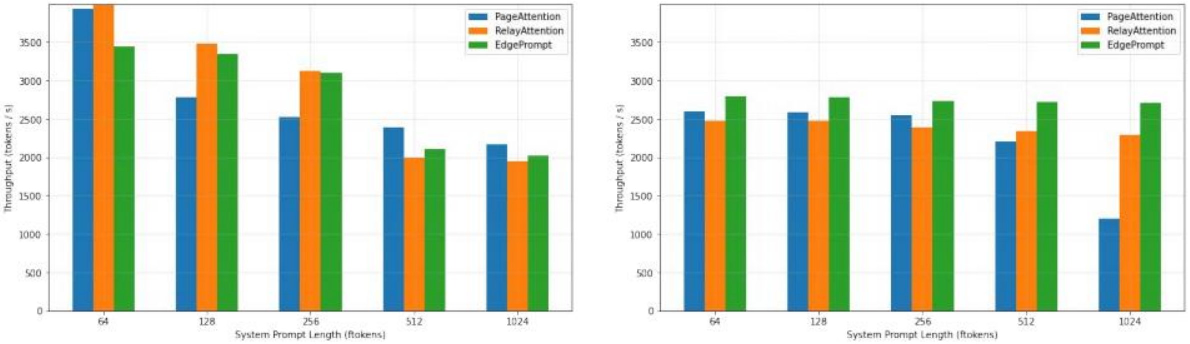
Batch Inference Results – High Throughput Under Load

Setup:

- 1,000 batched requests from ShareGPTv3Cloud
- prompt: 64–1024 tokens
- Edge prompt: fixed at 512 tokens

Throughput (Token/s) compared across:

- PageAttention
- RelayAttention
- EdgePrompt (ours)



(a) Servering on the Cloud

(b) Servering on the Edge

Key Observations:

- PageAttention drops sharply with long prompts
- RelayAttention stable but limited
- EdgePrompt achieves highest throughput, especially on edge device

Interactive Inference: Concurrency and Latency

Setup:

- 1,000 requests under Poisson-distributed arrival
- Cloud prompt: 512–1024 tokens

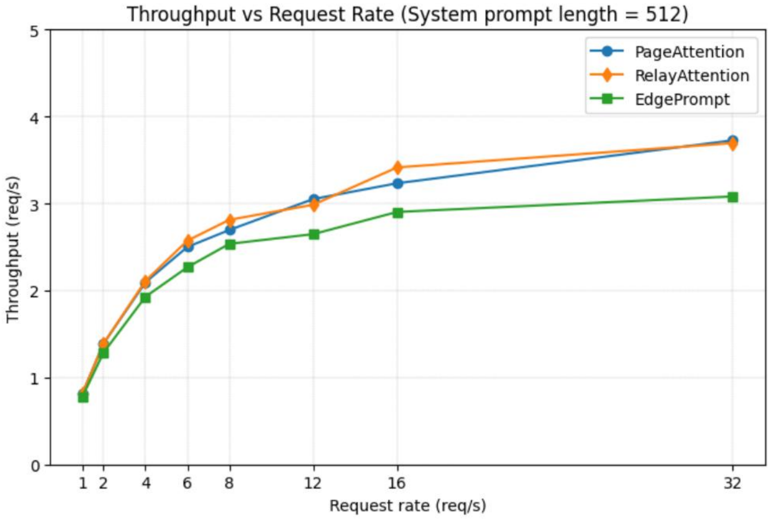
Metrics:

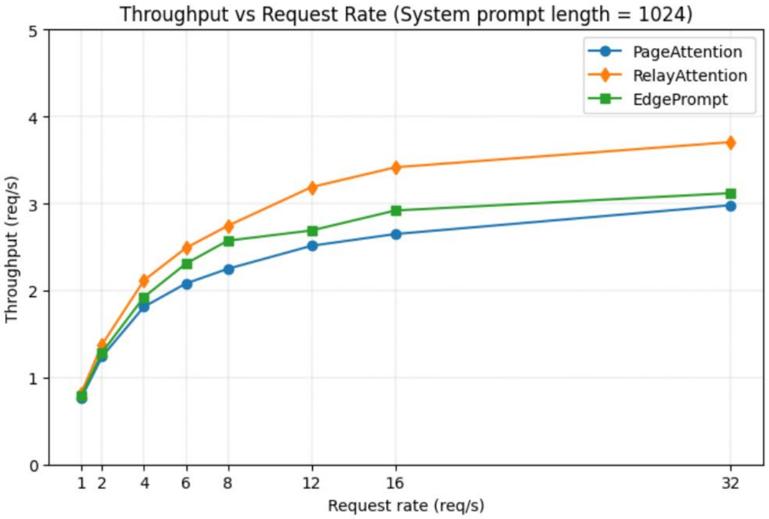
- Throughput (Requests/s) vs Request Rate
- Latency (Seconds/Token) under load

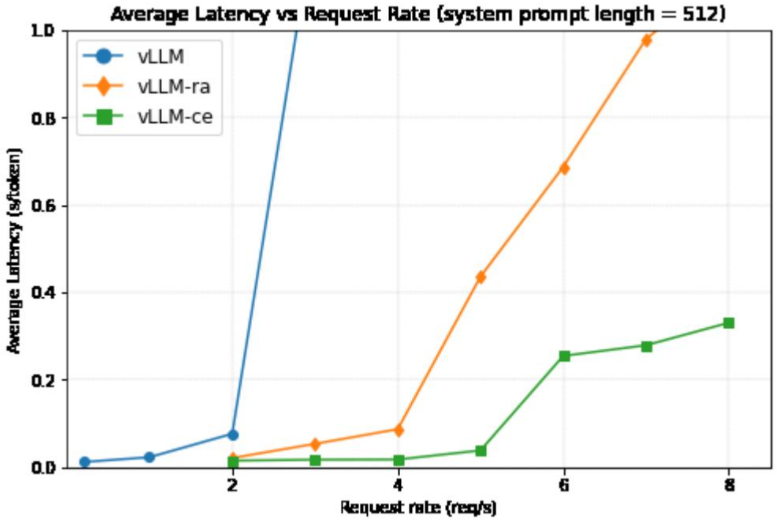
Key Observations:

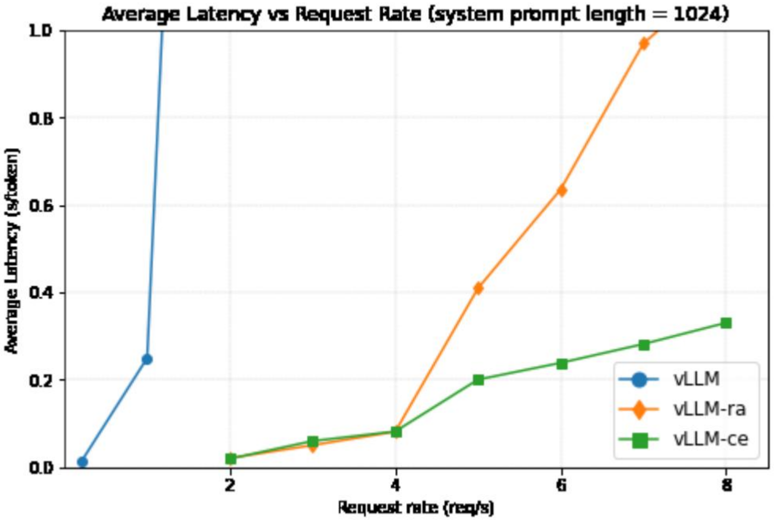
- EdgePrompt scales better with load
- Maintains lower latency growth
- Outperforms RelayAttention in high-concurrency conditions

Interactive Inference: Concurrency and Latency









Conclusion

- Introduced EdgePrompt: a cloud-edge collaborative inference framework for LLMs
- Separates cloud and edge prompts to optimize both privacy and efficiency
- Proposes KV synchronization + fused attention for distributed inferenc
- Models end-to-end latency with overlapped scheduling
- Experiments show higher throughput, lower latency, and better scalability
- Keeps user data local, ensuring privacy in 6G environments
- EdgePrompt is a scalable, practical solution for LLM inference in future networks

THANKS

04

PART

Experimental Results