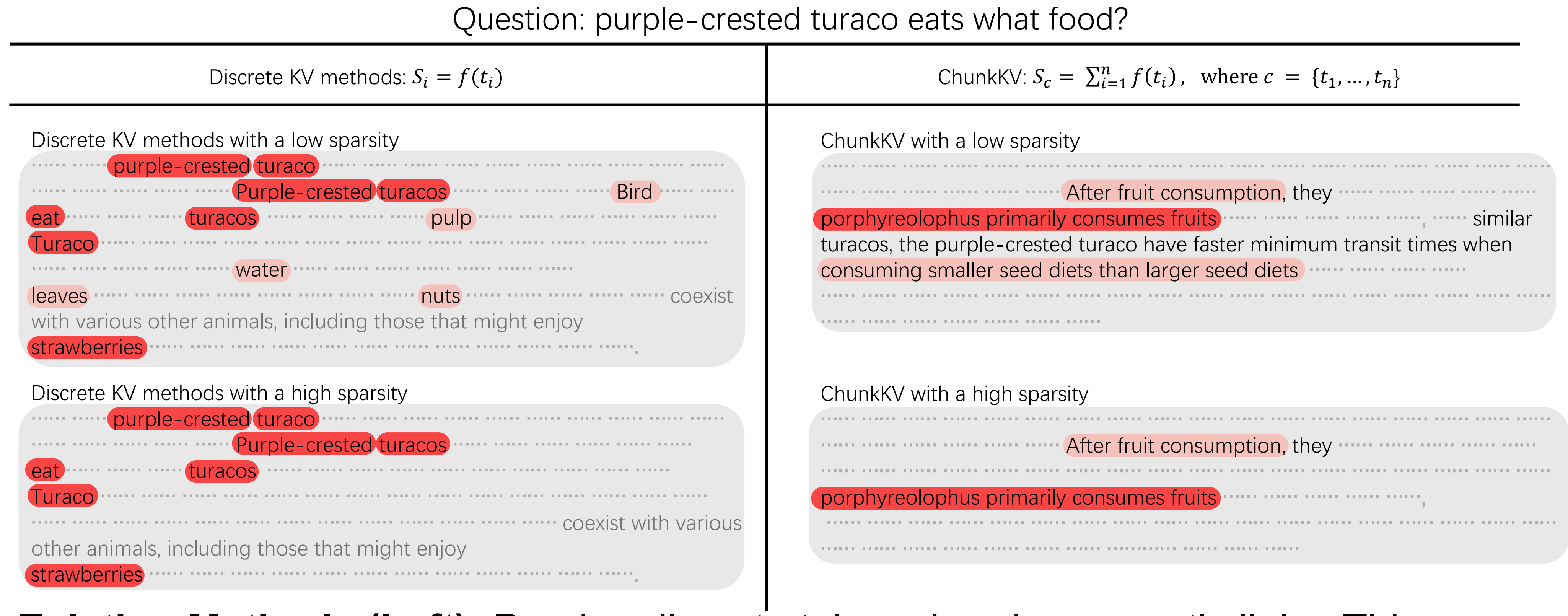
ChunkKV: Semantic-Preserving KV Cache Compression for Efficient Long-Context LLM Inference

Xiang Liu*, Zhenheng Tang*, Peijie Dong, Zeyu Li, Yue Liu, Bo Li,
Xuming Hu, Xiaowen Chu

Paper

WeChat

The Problem: Isolated Tokens vs. Semantic Chunks



Existing Methods (Left): Pruning discrete tokens breaks semantic links. This approach breaks semantic dependencies, leading to fragmented context and performance degradation.

ChunkKV (Right): Preserving semantic *chunks* retains the full context.

ChunkKV

We propose ChunkKV, which treats **semantic chunks**—not isolated tokens—as the basic unit of compression.

How it Works:

1.Group: Divide the KV Cache into fixed-size chunks (e.g., 10 tokens).

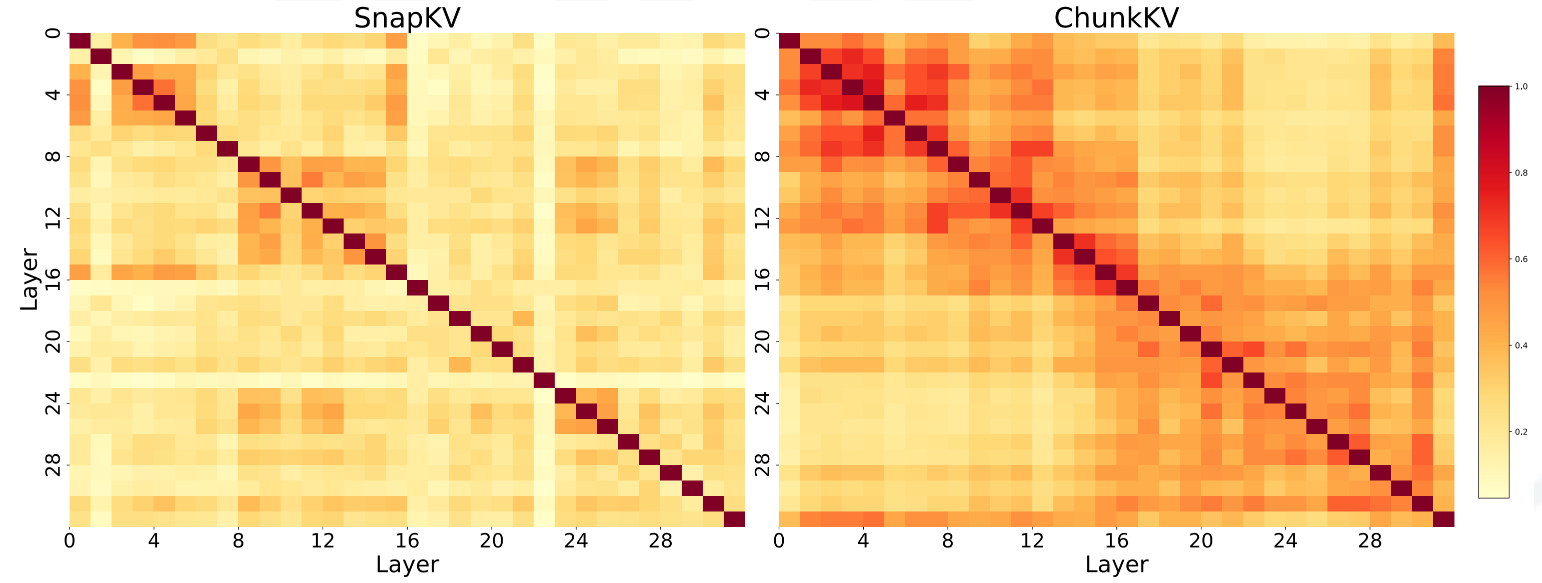
2.Score: Calculate each *chunk's* importance by summing the attention scores of its constituent tokens.

3.Select: Keep the Top-K most important chunks.

4.Preserve: Always keep the most recent tokens (the "Observe Window") to maintain context.

Layer-Wise Index Reuse

- **Key Insight:** We observed that the indices of important chunks are **highly similar across adjacent Transformer layers.**
- **The Technique:** We compute the important indices on layer L and **reuse** them for the next N layers.
- **The Benefit:** This dramatically reduces the computational overhead of the compression step itself.



ChunkKV (right) shows significantly higher inter-layer index similarity than SnapKV (left).

Performance

LongBench

Ratio	SLM	H2O	SKV	PKV	ChunkKV (Ours)
LlaMa-3-8B-Instruct FullKV: 41.46 ↑					
10%	-13.80%	-10.61%	-3.16%	-3.33%	**-2.29%**
20%	-6.42%	-8.85%	-2.24%	-2.00%	**-1.74%**
30%	-2.36%	-5.38%	-0.07%	-0.22%	**+0.31%**
Mistral-7B-Instruct-v0.3 FullKV: 48.08 ↑					
10%	-16.58%	-9.30%	-3.54%	-3.52%	**-2.85%**
Qwen2-7B-Instruct FullKV: 40.71 ↑					
10%	-5.28%	-0.64%	-0.39%	-0.98%	**+0.42%**
Qwen2-7B-Instruct on LongBench-ZH FullKV: 38.60 ↑					
10%	-15.95%	-5.31%	+0.18%	-5.31%	**+2.20%**

NIAH

KV cache Size	SLM	H2O	SKV	PKV	ChunkKV (Ours)
LlaMa-3.1-8B-Instruct FullKV: 74.6% ↑					
512	32.0%	68.6%	71.2 %	72.6%	**74.5%**
256	28.0%	61.7%	68.8%	69.5%	**74.1%**
128	23.7%	47.9%	58.9%	65.1%	**73.8%**
96	21.5%	41.0%	56.2%	63.2%	**70.3%**
Mistral-7B-Instruct FullKV: 99.8% ↑					
128	44.3%	88.2%	91.6%	99.3%	**99.8%**

GSM8K

Ratio	SLM	H2O	SKV	PKV	ChunkKV (Ours)
DeepSeek-R1-Distill-Llama-8B FullKV: 69.4% ↑					
10%	51.6%	55.6%	57.6%	62.6%	**65.7%**
LlaMa-3.1-8B-Instruct FullKV: 79.5% ↑					
30%	70.5%	72.2%	76.1%	77.1%	**77.3%**
20%	63.8%	64.0%	68.8%	71.4%	**77.6%**
10%	47.8%	45.0%	50.3%	48.2%	**65.7%**
LlaMa-3-8B-Instruct FullKV: 76.8% ↑					
30%	70.6%	73.6%	70.2%	68.2%	**74.6%**
Qwen2-7B-Instruct FullKV: 71.1% ↑					
30%	70.8%	61.2%	70.8%	64.7%	**73.5%**

Many-Shot GSM8K

Ratio	SLM	H2O	SKV	PKV	ChunkKV (Ours)
DeepSeek-R1-Distill-Llama-8B FullKV: 71.2% ↑					
10%	63.2%	54.2%	54.1%	59.2%	**68.2%**
LlaMa-3.1-8B-Instruct FullKV: 82.4% ↑					
10%	74.3%	51.2%	68.2%	70.3%	**79.3%**

SLM=StreamingLLM

SKV=SnapKV

PKV=PryamidKV

Index Reuse

Efficiency

Method	Sequence Length		Performance Metrics	
	Input	Output	Latency(s) ↓	Throughput(T/S) ↑
FullKV	4096	1024	43.60	105.92
ChunkKV	4096	1024	37.52 (13.9%)	118.85 (12.2%)
ChunkKV_reuse	4096	1024	**37.35** (14.3%)	**124.09** (17.2%)
FullKV	4096	4096	175.50	37.73
ChunkKV	4096	4096	164.55 (6.2%)	40.58 (7.6%)
ChunkKV_reuse	4096	4096	**162.85** (7.2%)	**41.12** (9.0%)
FullKV	8192	1024	46.48	184.08
ChunkKV	8192	1024	37.83 (18.6%)	228.96 (24.4%)
ChunkKV_reuse	8192	1024	**36.85** (20.7%)	**232.99** (26.5%)
FullKV	8192	4096	183.42	55.93
ChunkKV	8192	4096	164.78 (10.2%)	65.14 (16.5%)
ChunkKV_reuse	8192	4096	**162.15** (11.6%)	**66.05** (18.1%)

Performance

Model	ChunkKV	
	Baseline	Index Reuse$_{\triangle}$
LongBench		
LLaMA-3-8B-Inst	40.51	$40.27_{-0.59\%}$
Mistral-7B-Inst	46.71	$46.43_{-0.59\%}$
Qwen2-7B-Inst	40.88	$40.76_{-0.29\%}$
GSM8K		
LLaMA-3-8B-Inst	74.5	$74.6_{+0.13\%}$
Qwen2-7B-Inst	71.2	$71.2_{+0.00\%}$

Conclusion

1. **Preserves Semantics (SOTA Accuracy):** By retaining semantic chunks, ChunkKV outperforms SOTA methods on NIAH, LongBench, and GSM8K.
2. **Faster Inference (Better Efficiency):** Our Layer-Wise Index Reuse technique achieves up to **26.5% higher throughput** than the FullKV baseline.
3. **ChunkKV is a simple, effective solution that achieves both higher accuracy and faster inference speed.**

xliu886@connect.hkust-gz.edu.cn