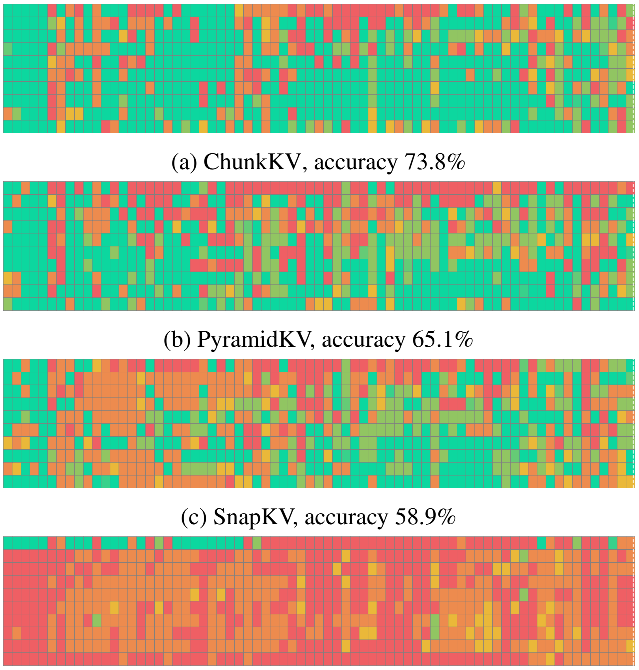
Table 7: NIAH Performance Comparison.

KV cache Size	StreamingLLM	H2O	SnapKV	PyramidKV	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%**

cates that the method can retrieve the needle at that length and depth percentage. The detail visualization of the NIAH benchmark can be found in Appendix B.3. The visualization results demonstrate that ChunkKV outperforms other KV cache compression methods.



(a) ChunkKV, accuracy 73.8%

(b) PyramidKV, accuracy 65.1%

(c) SnapKV, accuracy 58.9%

(d) StreamingLLM, accuracy 23.7%

Figure 3: NIAH benchmark for LLaMA3-8B-Instruct with KV cache size=128 under 8k context length.

4.3. Index Reuse

This section will evaluate the performance of the layer-wise index reuse approach with ChunkKV from the two aspects of efficiency and performance.

Measuring Efficiency. We evaluated the latency and throughput of ChunkKV compared to FullKV using LLaMA3-8B-Instruct on an A40 GPU. All experiments were conducted with reuse layer is 2, batch size set to 1 and inference was performed using Flash Attention 2, each experiment was repeated 10 times and the average latency and throughput were reported.

Table 8: Latency and throughput comparison between ChunkKV and FullKV under different input-output configurations. Percentages in parentheses indicate improvements over FullKV baseline.

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%)

The results in Table 8 shows that the layer-wise index reuse strategy (ChunkKV_reuse) further boosts performance, achieving up to a 20.7% reduction in latency, and throughput improvements are particularly notable for longer input sequences, with ChunkKV_reuse delivering up to a 26.5% improvement over FullKV.

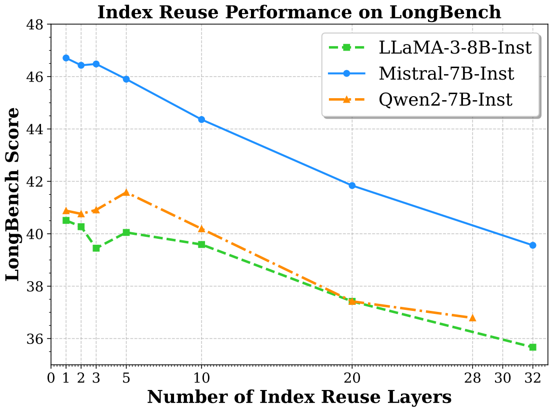


Figure 4: Comparison with different index reuse layers on LongBench.

Measuring Task Performance. This experiment evaluates the performance of the layer-wise index reuse approach by measuring the performance of the LongBench (Bai et al., 2024), the experiment settings are the same as LongBench in 4.2. And the number of index reuse layers is set from 1 to the number of layers in the model, where an index reuse layer of 1 corresponds to the normal ChunkKV without index reuse, and our method set reuse layer to 2.

Figure 4 illustrates the performance of ChunkKV with varying index reuse layers on the LongBench benchmark. Generally, reuse layer set to 2 can achieve the minimal performance degradation across all models. For more experiments