Table 4: Many-Shot GSM8K Performance Comparison.

Ratio	StreamingLLM	H2O	SnapKV	PyramidKV	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%**

For more details on the prompt settings, please refer to the APPENDIX G.

Table 3 presents the performance comparison. The results demonstrate that ChunkKV outperforms other KV cache compression methods on different models and compression ratios. Table 4 presents the performance comparison of many-shot GSM8K, also ChunkKV outperforms other KV cache compression methods. The consistent superior performance of ChunkKV in both models underscores its effectiveness in maintaining crucial contextual information for complex arithmetic reasoning tasks.

Jailbreak In this section, we evaluate the performance of ChunkKV on the JailbreakV benchmark (Luo et al., 2024). The prompt settings are the same as those used by Luo et al. (2024).

Table 5 presents the performance comparison. The results demonstrate that ChunkKV outperforms other KV cache compression methods on different models and compression ratios. Which shows the effectiveness of ChunkKV in maintaining crucial contextual information for safety benchmark.

Table 5: JailbreakV Performance Comparison.

Ratio	StreamingLLM	H2O	SnapKV	PyramidKV	ChunkKV (Ours)
LlaMa-3.1-8B-Instruct FullKV: 88.9% ↑					
20%	65.0%	71.7%	88.0%	87.5%	**89.0%**
10%	53.1%	65.4%	84.3%	85.5%	**87.9%**

4.2. Long-Context Benchmark

LongBench and NIAH are two widely used benchmarks for KV cache compression methods. Both benchmarks have a context length that exceeds $10K$. NIAH requires retrieval capability, while LongBench is a meticulously designed benchmark suite that tests the capabilities of language models in handling extended documents and complex information sequences.

LongBench We use LongBench (Bai et al., 2024) to assess the performance of ChunkKV on tasks involving long-context inputs. For more details on LongBench, please refer to the APPENDIX F. We evaluated multiple KV cache eviction methods using the LongBench benchmark with LLaMA-3-8B-Instruct (Meta, 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a), and Qwen2-7B-Instruct (Yang et al., 2024a), with a KV cache compression ratio of 10%. The LongBench provides the Chinese subtask, and Qwen2-7B-Instruct also supports Chinese, so we tested Qwen2-7B-Instruct with different KV cache compression methods on the Chinese subtasks.

Table 6: KV cache compression methods on the LongBench benchmark. Results show performance gap compared to FullKV baseline (negative values indicate worse performance).

Ratio	StreamingLLM	H2O	SnapKV	PyramidKV	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%**

Tables 6 show that ChunkKV is capable of achieving on-par performance or even better than the full KV cache with less GPU memory consumption. This table presents the performance gap (in percentage) between each method and the FullKV baseline, where negative values indicate performance degradation compared to FullKV. The table is evaluated in the LongBench English subtask, where ChunkKV outperforms other compression methods overall. This suggests that ChunkKV's approach of retaining semantic chunks is more effective in preserving important information compared to other discrete token-based compression methods. For detailed results and Chinese subtask results, please refer to Appendix B.2 and B.5.

Needle-In-A-HayStack We use Needle-In-A-HayStack (NIAH) (Kamradt, 2023) to evaluate LLMs' long-context retrieval capability. NIAH assesses how well LLM extract hidden tricked information from extensive documents, and follow LLM-as-a-Judge (Zheng et al., 2023) we apply GPT-4o-mini (OpenAI, 2023) to assess the accuracy of the retrieved information. We evaluated multiple KV cache eviction methods using NIAH with LLaMA-3-8B-Instruct and Mistral-7B-Instruct-v0.2, setting benchmark context lengths to 8k and 32k tokens.

Table 7 provides statistical results for different compression methods. These findings clearly indicate the effectiveness of ChunkKV in managing varying token lengths and depth percentages, making it a robust choice for KV cache management in LLMs. Figure 3 presents the NIAH benchmark results for LLaMA-3-8B-Instruct. The vertical axis represents the depth percentage, while the horizontal axis represents the token length, with shorter lengths on the left and longer lengths on the right. A cell highlighted in green indi-