Table 11: Detailed comparison of KV cache metrics across different task categories in LongBench.

Method	Single-Document QA	Multi-Document QA	Summarization	Few-shot Learning	Synthetic & Code
KV Cache L1 Loss ↓					
ChunkKV	**0.8741**	**0.8748**	**0.8770**	**0.8861**	**0.8726**
SnapKV	0.8921	0.8933	0.8930	0.8917	0.8938
H2O	0.8905	0.8917	0.8913	0.8906	0.8915
Attention Score Cosine Similarity ↑					
ChunkKV	**0.3567**	**0.3651**	**0.3841**	**0.4330**	**0.3805**
SnapKV	0.3513	0.3594	0.3771	0.4305	0.3759
H2O	0.3491	0.3572	0.3750	0.4284	0.3740

A.2. Hypothetical Scenario

To provide a deeper understanding of ChunkKV's effectiveness compared to discrete token-based methods, we present a detailed analysis using a hypothetical scenario. This analysis aims to illustrate the fundamental differences between these approaches and explain why ChunkKV is more effective at preserving semantic information in long contexts.

Consider a comprehensive document that contains detailed information on various animals, including their habitats, diets, and behaviors. A user asks the question "What do pandas eat in the wild?"

Both ChunkKV and discrete token-based methods would use this question to calculate observation scores for the document. However, their approaches to selecting and retaining information differ significantly.

A.2.1. Discrete Token-based Method

A discrete token-based method might identify and retain individual tokens with high relevance scores, such as:

- "pandas","eat", "bamboo", "wild", "diet", "food"

Although these tokens are relevant, they lack context and coherence. The method might discard other essential tokens that provide crucial context or complete the information.

A.2.2. ChunkKV Method

In contrast, ChunkKV would identify and retain semantically meaningful chunks, such as:

- "In the wild, pandas primarily eat bamboo shoots and leaves"
- "Their diet consists of 99% bamboo, but they occasionally consume other vegetation"
- "Wild pandas may also eat small rodents or birds when available"

By preserving these chunks, ChunkKV maintains not only the relevant keywords but also their contextual relationships and additional pertinent information.

A.3. Comparative Analysis

The advantages of ChunkKV become evident when we consider how these retained pieces of information would be used in subsequent processing:

1. **Contextual Understanding**: Discrete tokens require the model to reconstruct meaning from isolated words, which could lead to ambiguity. ChunkKV provides complete phrases or sentences, allowing for immediate and accurate comprehension.