reasoning (O1 and R1) LLMs, achieving state-of-the-art performance.

2. Related Work

KV Cache Compression. KV cache compression technology has developed rapidly in the era of LLM, with methods mainly focused on evicting unimportant tokens. The compression process occurs before the attention blocks, optimizing both the prefilling time and GPU memory. Xiao et al. (2024) and Han et al. (2024) propose that initial and recent tokens consistently have high attention scores between different layers and attention heads. As a result, retaining these tokens in the KV cache is more likely to preserve important information. Furthermore, FastGen (Ge et al., 2023) evicts tokens based on observed patterns. H2O (Zhang et al., 2023) and SnapKV (Li et al., 2024) employ dynamic KV cache compression methods, evaluating the importance of tokens based on attention scores and then evicting the less important ones. As inference scenarios become increasingly complex, dynamic KV cache compression methods demonstrate powerful performance. Recently, Yang et al. (2024b) and Cai et al. (2024) have closely examined the distributions of attention scores during the pre-filling stage of the Retrieval-Augmented Generation (RAG) task, discovering a pyramidal KV cache compression pattern in different transformer layers.

Although these KV cache compression methods have explored efficient GPU memory management while maintaining original performance, our study focuses more on the semantic information of the prompt. We find that chunks of the original KV cache are more important than discrete tokens.

Chunking Method. The chunking methodology is widely used in the field of NLP due to its simplicity and effectiveness (Tjong Kim Sang & Veenstra, 1999). In the era of LLMs, chunking is primarily applied in data pre-processing. For example, Shi et al. suggest grouping related training data into chunks to achieve better convergence curves to pre-train LLMs. Fei et al. (2024) apply a topic-based chunking method to improve the semantic compression of prompts. Furthermore, chunking plays an important role in the Retrieval-Augmented Generation (RAG) field (Yepes et al., 2024; Smith & Troynikov, 2024; Anthropic, 2024). It serves to divide documents into units of information with semantic content suitable for embedding-based retrieval and processing by LLMs.

Layer-Wise Technique The layer-wise technique is widely used in the training and inference of large language models (LLMs). LISA (Pan et al., 2024a) is a layer-wise sampling method based on observations of the training dynamics of Low-Rank Adaptation (LoRA)(Hu et al., 2022) across layers. LAMB(You et al., 2020) is a layer-wise adaptive learning rate method that speeds up LLM training by stabilizing training convergence with large batch sizes. DoLa (Chuang et al., 2023) employs layer-wise contrasting to reduce hallucinations during LLM inference.

3. ChunkKV

3.1. Preliminary Study of KV Cache Compression

With the increasing long-context capabilities of LLMs, the KV cache has become crucial for improving inference efficiency. However, it can consume significant GPU memory when handling long input contexts. The GPU memory cost of the KV cache for the decoding stage can be calculated as follows:

$$M_{KV} = 2 \times B \times S \times L \times N \times D \times 2 \tag{1}$$

where B is the batch size, S is the sequence length of prompt and decoded length, L is the number of layers, N is the number of attention heads, D is the dimension of each attention head, and the first 2 accounts for the KV matrices, while the last 2 accounts for the precision when using float16. Table E shows the configuration parameters for LLaMA-3-8B-Instruct (Meta, 2024). With a batch size $B = 1$ and a sequence length of prompt $S = 2048$, the GPU memory cost of the KV cache is nearly 1 GB. If the batch size exceeds 24, the GPU memory cost of the KV cache will exceed the capacity of an RTX 4090 GPU. To address this issue, KV cache compression methods have been proposed, with the aim of retaining only a minimal amount of KV cache while preserving as much information as possible. For more details on the LLM configuration parameters, refer to Appendix E.

3.2. Chunk Based KV Compression

To address the limitations of existing KV cache compression methods, we propose ChunkKV, a novel KV cache compression method that retains the most informative semantic chunks. The key idea behind ChunkKV is to group tokens in the KV cache into chunks that preserve more semantic information, such as a chunk containing a subject, verb and object. As illustrated in Figure 1, ChunkKV preserves the chunks of the KV cache that contain more semantic information. First, we define a chunk as a group of tokens that contain related semantic information. By retaining the most informative chunks from the original KV cache, ChunkKV can effectively reduce the memory usage of the KV cache while preserving essential information.

The Algorithm 1 shows the pseudocode outline of ChunkKV. First, following H2O (Zhang et al., 2023) and SnapKV (Li et al., 2024), we set the observe window by computing the observation scores $\mathbf{A} \leftarrow \mathbf{Q}_{T_q-w:T_q}\mathbf{K}^T$, where $\mathbf{Q}_{T_q-w:T_q}$