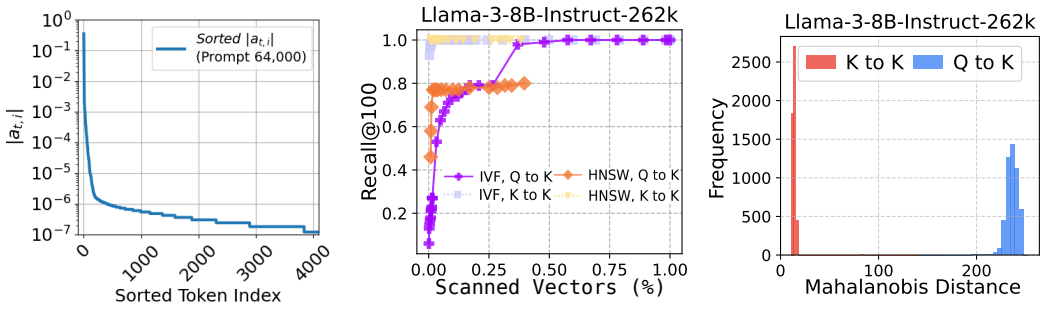

(a) Attention score distribution. (b) ANNS index performance. (c) Different distribution.

Figure 2: (a) Attention is highly sparse in 64,000 tokens. (b) Off-the-shelf ANNS indexes perform poorly on Q to K searches but performs well on K to K searches. The Q and K are dumped from Llama-3-8B with a prompt length of 128,000 tokens. (c) Query vectors (Q) are distant from key vectors (K) while key vectors (K) themselves are close.

2 Background and Motivation

2.1 LLM and Attention Operation

In the generation of the t-th token, the attention mechanism computes the dot product between the query vector $\mathbf{q}_t \in \mathbb{R}^{1\times d}$ (where d is the hidden dimension) and the key vector of past tokens $\mathbf{k}_i \in \mathbb{R}^{1\times d}$ (for $i \leq t$). This product is scaled by $d^{-\frac{1}{2}}$ and normalized via a `Softmax` function to yield the attention score $a_{t,i}$. These scores then weight the values $\mathbf{v}_i$, resulting in the output $\mathbf{o}_t$.

$$z_i = \frac{\mathbf{q}_t \cdot \mathbf{k}_i^T}{\sqrt{d}}, \quad a_{t,i} = \frac{e^{z_i}}{\sum_{j=1..t} e^{z_j}}, \quad \mathbf{o}_t = \sum_{i=1..t} a_{t,i} \cdot \mathbf{v}_i \tag{1}$$

LLM inference contains two stages: the prefill phase and decoding phase. The prefill phase, which only happens once, computes the keys and values of the prompt with a time-complexity $O(n^2)$. In the decoding (token generation) phase, the newly generated token becomes the new query and computes attention scores with all past key vectors. One common optimization to avoid repetitive calculation is to cache past KV states in the GPU memory, thereby reducing the complexity to $O(n)$.

2.2 Expensive Long-Context Serving

Due to the quadratic time complexity of attention operation, serving long-sequence input incurs extremely high cost. Table 1 shows the inference latency without KV cache. When the prompt length reaches 1 million tokens, generating every token requires 1,765 seconds with over 96% of latency spent on attention operations. Although KV cache can reduce the decoding latency, it demands a huge amount of GPU memory for long contexts. As shown in Table 1, 500 GB memory is necessary for storing the KV cache when the context length reaches 1 million tokens, which is far beyond the GPU memory capacity of a single A100 GPU (80 GB). Offloading and reloading the KV cache between GPU and CPU memory is a potential solution but incurs excessive commutation overhead over PCIe [4], degrading the inference performance especially on commodity GPUs.

2.3 Dynamic and Sparse Attention

Despite the large size of the context, only a small proportion of tokens actually dominate the generation accuracy. Figure 2a shows the distribution of $|a_{t,i}|$ in Equation 1 for a query vector from Llama-2-7B with a prompt of 64,000 tokens. We observe that the top 500 tokens dominate the values of $|a_{t,i}|$, while the remaining tokens contribute approximately zero. A high attention score indicates that two vectors are close, as measured by the inner product. Therefore, the sparsity of attention scores means that the relevant keys to the query are very few. We measure the mean-square-error (MSE) of the attention output if we use the top-k $|a_{t,i}|$ as an approximation. We find that it only