Table 17: Performance comparison of Chinese subtask on LongBench for Qwen2-7B-Instruct.

| Method | Single-Document QA | Multi-Document QA | Summarization | Few-shot Learning | Synthetic | Avg. ↑ |
|---|---|---|---|---|---|---|
| | MF-zh | DuReader | VCSum | LSHT | PR-zh | |
| Avg len | 6,701 | 15,768 | 15,380 | 22,337 | 6,745 | |
| Qwen2-7B-Instruct, KV Size = Full | | | | | | |
| FullKV | 39.17 | 23.63 | 16.21 | 43.50 | 70.50 | 38.60 |
| Qwen2-7B-Instruct, KV Size Compression Ratio = 10% | | | | | | |
| StreamingLLM | 38.05 | **23.24** | 15.92 | 40.50 | 44.50 | 32.44 |
| H2O | 37.99 | 19.58 | 16.16 | 41.67 | 67.35 | 36.55 |
| SnapKV | 44.25 | 20.27 | 16.24 | **44.50** | 68.10 | 38.67 |
| PyramidKV | 36.57 | 20.56 | 16.15 | 43.50 | 66.50 | 36.55 |
| **ChunkKV** | **45.92** | 20.15 | **16.37** | 43.75 | **71.10** | **39.45** |

## C. Theoretical Understanding

In this section, we provide the theoretical interpretation from the perspective from the In-context learning (ICL) to further understand how ChunkKV outperforms token-level KV cache compression.

**Pretraining Data Distribution.** Given a set of concepts $\Theta$ and a concept $\theta \in \Theta$, we define the pretraining data is sampled from $p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T|\theta)p(\theta)d\theta$ (Fang & Xie, 2022). Each token $o$ is sampled from a vocabulary $\mathbb{O}$. For simplicity, we write $o_{1:t} = o_1 \ldots o_t$.

**Language Modeling.** Current LLMs (Brown et al., 2020; Touvron et al., 2023; Fang & Xie, 2022) usually utilize the next word prediction as the language modelling, which predicts the next token $o_t$ given the previous tokens $o_1 \ldots o_{t-1}$ for all $t = 1, \ldots, T$. Formally, a language modelling can be writen as the distribution $f(o_t|o_{1:t-1})$. And it is pretrained on a huge corpus sampled from the pretraining distribution $p(o_1, \ldots, o_{t+1})$ (Fang & Xie, 2022). Considering the large scale of the model size and dataset size, it can be assumed that the $f(o_1 \ldots o_{t+1})$ has been aligned with the $p(o_1 \ldots o_{t+1})$ (Fang & Xie, 2022).

**Prompt Distribution.** Following (Fang & Xie, 2022), a prompt is composed of an input token sequence $x$ followed by an output token $y$. Then, the $i$-th training example [1] that can appear in any place in the whole prompt $o_{1:T}$ is defined as $O_i$ consisting of an input $x_i = O_i\,[1 : k-1]$ (the first $k-1$ tokens) followed by the output $y_i = O_i\,[k]$ at the end, where the length $k$ is fixed for simplicity.

The $i$-th training example is independently generated as follows: 1) Generate a start hidden state $h_i^{\text{start}}$ from a *prompt start distribution* $p_{\text{prompt}}$; 2) Given $h_i^{\text{start}}$, generate the example sequence $O_i = [x_i, y_i]$ from $p(O_i|h_i^{\text{start}}, \theta^\star)$. The test input $x_{\text{test}} = x_{n+1}$ is sampled similarly. Then, the prompt consists of a sequence of training examples ($S_n$) followed by the example $x_{\text{test}}$:

$$[S_n, x_{\text{test}}] = [x_1, y_1, x_2, y_2, \ldots, x_n, y_n, x_{\text{test}}] \sim p_{\text{prompt}}. \tag{2}$$

**In-context learning setups and Assumptions.** We follow other settings and assumptions in (Fang & Xie, 2022). With the greedy decoding (Fu et al., 2024b), sampling the next token from the language modeling $f(o_t|o_{1:t-1})$ becomes the predictor as $y = \arg\max_{o_t} f(o_t|o_{1:t-1})$.

Thus, for $[S_n, x_{\text{test}}]$, the in-context learning predictor can be written as $f_n(x_{\text{test}}) := \arg\max_y p(y|S_n, x_{\text{test}})$, which outputs the most likely prediction over the *pretraining distribution* conditioned on the *prompt distribution*. Its expected 0-1 error with $n$ examples is $L_{\text{0-1}}(f_n) = \mathbb{E}_{x_{\text{test}}, y_{\text{test}} \sim p_{\text{prompt}}}[\mathbf{1}[f_n(x_{\text{test}}) \neq y_{\text{test}}]]$.

We define $p_\theta^i(o) := p(O[i] = o|O[1 : i-1], \theta)$ of the $i$-th token with previous tokens and the analogous distribution $p_{prompt}^i := p_{prompt}(O[i] = o|O[1 : i-1])$ under the prompt distribution. Following (Fang & Xie, 2022), there is a distinguishability condition formalizes when in-context learning occurs giving the concept $\theta$.

The distinguishability condition is dependent on a KL divergence between the previous two distributions and the error terms

[1] Here, training example in prompts means happens during the prompt learning.