Cross-Self KV Cache Pruning for Efficient Vision-Language Inference

Xiaohuan Pei, Tao Huang, Chang Xu
{xiaohuan.pei,t.huang,c.xu}@sydney.edu.au
The University of Sydney
Sydney, Australia

Abstract

KV cache pruning has emerged as a promising technique for reducing memory and computation costs in long-context auto-regressive generation. Existing methods for vision-language models (VLMs) typically rely on self-attention scores from large language models (LLMs) to identify and prune irrelevant tokens. However, these approaches overlook the inherent distributional discrepancies between modalities, often leading to inaccurate token importance estimation and the over-pruning of critical visual tokens. To address this, we propose decomposing attention scores into intra-modality attention (within the same modality) and inter-modality attention (across modalities), enabling more precise KV cache pruning by independently managing these distinct attention types. Additionally, we introduce an n-softmax function to counteract distribution shifts caused by pruning, preserving the original smoothness of attention scores and ensuring stable performance. Our final method, **C**ross-**S**elf **P**runing (CSP), achieves competitive performance compared to models with full KV caches while significantly outperforming previous pruning methods. Extensive evaluations on MileBench, a benchmark encompassing 29 multimodal datasets, demonstrate CSP's effectiveness, achieving up to a 41% performance improvement on challenging tasks like conversational embodied dialogue while reducing the KV cache budget by 13.6%. The code will be released.

Keywords

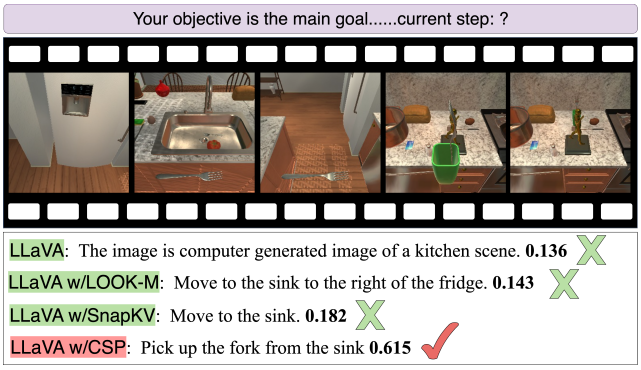
Vision Language Inference, KV Cache, Multi-Modality

ACM Reference Format:
Xiaohuan Pei, Tao Huang, Chang Xu. 2018. Cross-Self KV Cache Pruning for Efficient Vision-Language Inference. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

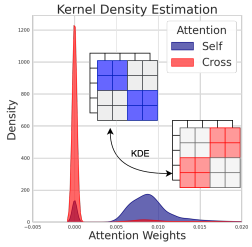
1 Introduction

The success of large language models (LLMs) [1, 4, 6, 38, 41, 44] has propelled the advancement of large vision-language models (VLMs) [5, 11, 22, 24, 26, 37, 42], enabling powerful integration and reasoning over multimodal inputs that combine both text and visual tokens. Unlike single-modal contexts, multimodal samples often comprise numerous images alongside text instructions, creating extended context lengths that challenge efficient inference.

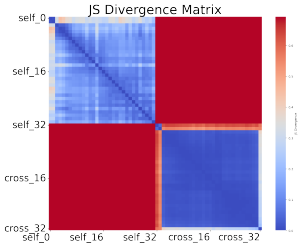
To address the challenges of long-context generation, KV caching has become a standard technique, where previously computed keys and values in the attention layers are stored in memory for reuse during subsequent generation steps. However, this approach still faces significant memory limitations, particularly for GPU and machine memory constraints. Recent works [8, 20, 23, 31, 46] have explored pruning unimportant tokens within the KV cache to alleviate memory demands, primarily by leveraging attention scores. Methods such as SnapKV [20] and H2O [46] apply this strategy to vision-language modeling (VLM) tasks by treating visual and text tokens uniformly across long sequences during pruning. Unfortunately, these methods rely on original attention scores that



(a) The generated response of a sample on the ALFRED [33].



(b) KDE for Attention Map



(c) Layer-Wise JS Estimations

Figure 1: (a) Performance improvement on the conversational embodied dialogue task. Figures (b) and (c) illustrate the distribution gap between self-attention and cross-attention during the decoding process in VLM tasks: (b) shows the Kernel Density Estimation (KDE) of attention weight distributions, and (c) presents the Jensen-Shannon (JS) divergence scores between cross-attention and self-attention across all layers.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

mix different modalities, potentially leading to suboptimal pruning outcomes.

In this paper, we identify a critical limitation in previous KV cache pruning methods: the distributional discrepancy between visual and textual modalities leads to inaccurate token importance estimation. Specifically, as illustrated in Figure 1b, self-attention scores (within a single modality) and cross-attention scores (across modalities) exhibit distinct and non-overlapping distributions. This divergence highlights that each attention type captures unique aspects of the input space, reflecting modality-specific priorities during decoding. Furthermore, as shown in Figure 1c, the Jensen-Shannon (JS) divergence between cross-attention and self-attention distributions reveals substantial variation across layers in LLaVA-7b. Relying solely on these mixed distributions for pruning introduces a selection bias: the pruned tokens tend to cluster within a single region or modality. This imbalance disrupts cross-modal interactions, ultimately degrading the model's performance.

Inspired by these observations, we hypothesize that independently selecting important tokens from distinct distributions can provide a more balanced and effective pruning strategy. To achieve this, we propose decomposing the attention matrix into intra-modality attention (within the same modality) and inter-modality attention (across different modalities), and performing token selection independently within each category. This approach ensures accurate preservation of critical tokens from both modalities while maintaining the efficiency and simplicity of prior methods. Additionally, we observe that pruning inherently shifts the distribution of attention scores, leading to degraded performance. To mitigate this, we introduce an n-softmax function that smooths the post-pruning distribution, effectively restoring the original smoothness of the attention scores and improving overall performance.

Our final method, **C**ross-**S**elf **P**runing (CSP), effectively addresses the challenges of KV cache pruning by leveraging independent intra-modality and inter-modality token selection along with n-softmax smoothing. CSP achieves a superior balance between performance and memory efficiency, significantly reducing the KV cache size while preserving or even enhancing model accuracy. We conduct extensive experiments on various VLMs, including LLaVA-v1.5-7b [24], InternVL [11], and MobileVLM [12]. The results demonstrate that CSP consistently outperforms existing methods like SnapKV [20] and H2O [46], achieving up to a 41% improvement in performance on complex tasks such as conversational embodied dialogue [33] , while reducing the KV cache budget by up to 13.6%.

2 Related Work

2.1 Vision-language models

Following the remarkable success of large language models (LLMs) [1, 4, 6, 38, 41, 44], recent research has focused on generative large vision-language models (VLMs) [5, 11, 22, 24, 26, 37, 42] to enhance multimodal comprehension and generation by leveraging the generalization capabilities of LLMs. Using the multi-modal pre-trained visual foundation models such as CLIP [30] as the visual encoder, existing methods commonly utilize extensive image-text data to align the visual encoder with LLMs, enabling the LLM to process and interpret visual inputs. For example, Flamingo [3] incorporates visual features into the LLM through gated attention, while LLaVA [26] connects the vision encoder and LLM via MLPs, demonstrating strong performance in multimodal dialogues.

2.2 KV cache optimization

Recent advancements in large language models (LLMs) have achieved notable success in optimizing KV cache for efficient long-context processing. Existing work on KV cache optimization [2, 8, 14, 20, 23, 31, 46] primarily utilizes attention scores to retain important tokens and improve memory efficiency in long-context processing. For example, SnapKV [20] introduces a technique for identifying attention allocation patterns and compressing the KV cache by pooling key tokens. H2O [46] presents a dynamic eviction policy that balances recent and frequently accessed tokens, identifying heavy hitters based on attention scores alone. ReCo [31] improves existing eviction strategies through refined importance scoring and structured eviction scopes. Keyformer [2] introduces gumbel softmax to relieve the impact of pruning tokens. More recent works [39, 45] have shifted towards adapting KV cache inference to multimodal contexts, where modality-specific properties are considered. For instance, LOOK-M [39] applies modality awareness to selectively evict image tokens and merge them with text tokens, prioritizing the retention of text-based KV pairs.

3 Method

Our approach decomposes the attention scores into intra-modality and inter-modality attention. We apply top-k selection within each region, ranking the tokens by summing attention scores along the query dimension for each key.

3.1 Preliminary

The auto-regressive generation of text yields a multi-step generation process. At each step i, the current token x_i is predicted by the LLM based on the input of prompt and previously-generated tokens $\{x_j\}_{j=1}^{i-1}$.

For reducing the computational cost and avoiding duplicated computations, current inference usually adopts KV cache technique, where the keys (K_j) and values (V_j) in self-attention of each previous tokens x_j are cached in memory and reused in subsequent steps.

However, when the context length is long, the storage of KV values still poses significant challenges in memory size and memory access speed. Therefore, to reduce the memory cost and run LLMs on resource-limited devices, KV cache pruning methods[2, 8, 14, 20, 23, 31, 46] are proposed to remove the less-important tokens.

Generally, the method contains two main components: (1) an importance estimation function f to measure the importance of each token x_j; (2) a selecting strategy to keep important tokens in KV and remove the rest based on their importance. Formally, with KV sequences to be cached, and T denotes the maximum cache length, it first measures the importance scores of each token using f, then generate the selective mask $M, M_i \in \{0, 1\}, \forall i = \{1, \ldots, T\}$ to select the tokens with top-T highest scores, where the tokens with zero values in M are omitted.

3.2 Cross-self pruning

Previous KV cache pruning methods [8, 20, 23, 31, 39, 46] usually use the self-attention scores to indicate the importance of each

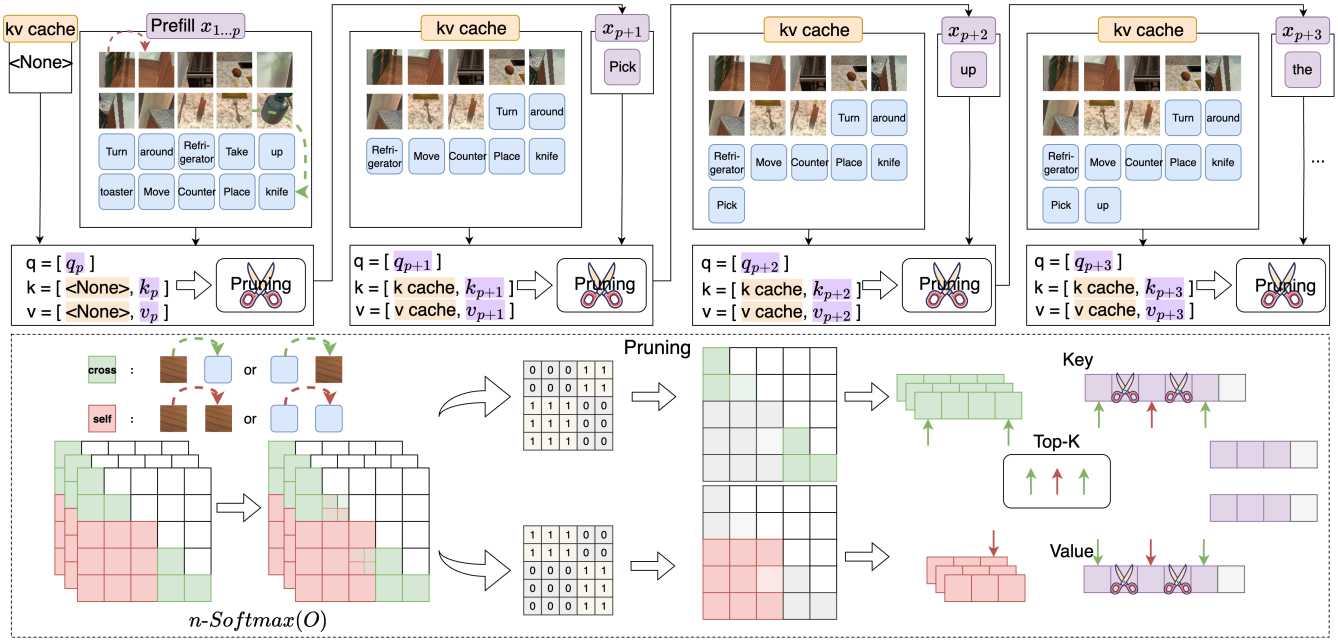


Figure 2: This illustration depicts the Cross-Self Pruning (CSP) KV cache process. The input sequence with vision and language tokens is projected into query and key representations across multiple modalities. The n-softmax attention weights serve as the selection function, which is decomposed into intra- and cross-modality. Summation is performed along the query axis within each region, and top-k keys are selected along the key dimension to retain tokens for pruning. ■ is the cached keys/values, and the ■ is current generated tokens. The concatenated cache tokens and recent tokens are forwarded to the CSP mechanism for pruning once the cache budget is reached.

token, as the attention scores determine the contributions of tokens to the attention output. Nevertheless, some recent works [39, 45] simply adopt the same strategy for pruning vision-language hybrid tokens, neglecting the distribution gap between different modalities. As validated in Figure 1, the discrepancy between attention scores within the same modality and different modalities are significant, resulting in overestimate or underestimate of the importance when considering both modalities together[1].

Motivated by this, for a more accurate estimation for VLMs, we aim to decompose the attention scores into two parts: intra-modality attention and inter-modality attention. The intra-modality attention denotes the attention scores between tokens within the same modality itself, and inter-modality attention is the ones across different modalities.

Mathematically, for an attention matrix $A \in [0,1]^{L\times L}$ (we average over the head axis if there are multiple heads in attention), where $L = L_t + L_v$ is the text-visual sequence length, we denote $A^{st} \in [0,1]^{L_t\times L_t}$ and $A^{sv} \in [0,1]^{L_v\times L_v}$ as the self-attention scores between text tokens and visual tokens, $A^{ct} \in [0,1]^{L_v\times L_t}$ and $A^{cv} \in [0,1]^{L_t\times L_v}$ as the visual-text (text as key) and text-visual cross-attention scores, respectively. Then, we can sum over the query axis to indicate the intra and inter importance of all the keys:

$$A^s = \sum_{k=1}^{L_t} A_k^{st} \oplus \sum_{k=1}^{L_v} A_k^{sv}, \quad A^c = \sum_{k=1}^{L_t} A_k^{ct} \oplus \sum_{k=1}^{L_v} A_k^{cv}, \tag{1}$$

where $\oplus$ denotes concatenation. Then the selective masks $M^{(s)}$ and $M^{(c)}$ of each matrix A^* are obtained by the top-K of the scores:

$$M^* = \{M_i\}_{i=1}^{L}, \quad \text{with } M_i^* = \begin{cases} 0, & i \notin \text{top-K}(A^*) \\ 1, & i \in \text{top-K}(A^*) \end{cases}. \tag{2}$$

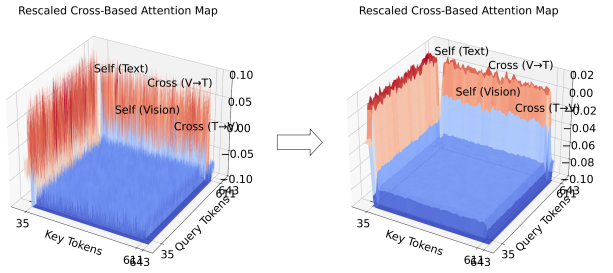
We set different K values K^s and K^v for intra attention and inter attention, which we find better and will be discussed in experiment section.

Finally, the mask for selecting tokens is from the tokens both important and selected by M^s and M^c:

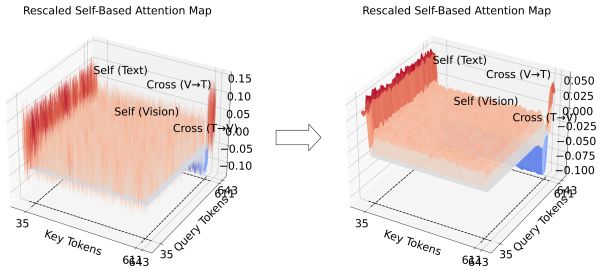
$$M = M^s \wedge M^c. \tag{3}$$

Also note that for better efficiency and accuracy, we trim the attention matrix by a number O of the most recent query tokens as an observation window and a number R of the previous key tokens, *i.e.*, $A[-O:,:-R]$, to reflect the actual needs in the generation of recent context. Figure 1 presents the cross-self selection approach. The green part refers to intra-modality, and the orange part to cross-modality. For both regions, we sum along the query axis, rank the score indices, and treat these as the selected key and value indices to retain. Due to potential overlap between the selected candidates M^s and M^c, the cache budget is significantly optimized across most

[1]Specifically, we find that on VLMs such as LLaVA, the attention scores of text tokens are usually larger than that of visual tokens, potentially leading to the loss of important visual information after KV pruning.



(a) n-Softmax applied on Cross-based Attention Map



(b) n-Softmax applied on Self-based Attention Map

Figure 3: Attention score smoothness. Figure (a) presents a case with high cross-attention activation, reflecting strong inter-modal dependency. Figure (b) shows a case with dominant self-attention, indicating strong intra-modal reliability. Both are smoothed using n-Softmax to prepare for modality-aware pruning within the cache.

modality scenarios, resulting in substantial savings in KV cache space. Finally, the combination of cross-self pruned tokens with recent tokens constitutes the optimized KV cache tokens:

$$\begin{aligned} K &= (K \odot M) \oplus K[-R:] \\ V &= (V \odot M) \oplus V[-R:], \end{aligned} \tag{4}$$

where $\odot$ is element wise operation. The pruned key and value are stored in the cache for later inference decoding. Algorithm 1 presents the cross-self decoding procedure during inference.

3.3 Smoothness recovery of attention scores

By using KV cache pruning, we can obtain reduced KV sequences for smaller costs. However, we find there is a sharpness-shift issue in the new pruned attention scores. Let us first consider the original computation of attention scores:

$$A = softmax(O), \quad \text{with } O = (\frac{QK^T}{\sqrt{d}}), \tag{5}$$

where d is a factor for stabilizing the values.

Comparing the attention score A_i between original and post-pruning ones, the difference occurs in the denominator of $softmax$, *i.e.*,

$$\frac{e^{(O_i)}}{\sum_{j\in I^+} e^{(O_j)} + \sum_{j\in I^-} e^{(O_j)}} \rightarrow \frac{e^{(O_i)}}{\sum_{j\in I^+} e^{(O_j)}}, \tag{6}$$

where I^+ and I^- denote the indices of tokens to be kept and pruned, respectively. It is clear to see that the A_i before pruning (left)is

Algorithm 1 Cross-**S**elf **P**runing (CSP) Procedure.

Input: $O \in \mathbb{R}^{H\times L_q\times L_k}$, current keys and values in the cache K, Q, budget T, recent size R, observation window O
for iteration $i \leftarrow 1$ to N **do**
  $L_k \leftarrow K$ ▷ Get key sequence length.
  if $L_k < T$ **then**:
    Return: K, V
  end if
  $A \leftarrow n\text{-}Softmax(O)$ ▷ Select function (Eq. 7).
  $A^{st}, A^{sv}, A^{ct}, A^{cv} \leftarrow A$ ▷ Mask of image and text
  $A^s, A^c \leftarrow A^{st}, A^{sv}, A^{ct}, A^{cv}$
  ▷ Acquire cross-attention and intra-attention (Eq. 1).
  $M^s \leftarrow \text{Topk}(A^s)$, $M^c \leftarrow \text{Topk}(A^c)$
  ▷ Get Top-K masks from intra- and cross-modality.
  $M \leftarrow M^s \wedge M^c$ ▷ Tokens that important both from intra- and cross-modality
  $K = (K \odot M) \oplus K[-R:]$,
  $V = (V \odot M) \oplus V[-R:]$
  ▷ Concatenate pruned tokens and recent tokens.
  Return: K, V ▷ Pruned Keys and Values
end for

smaller than the one after pruning (right), indicating that the original A is smoother. This changes would affect the attention outputs by overly enlarging the contributions of tokens with high attention scores, and therefore weaken the performance.

To address this issue, we propose a simple and effective function to recover the original smoothness of attention scores, which we call *n-softmax*, and A_i becomes:

$$A_i = n\text{-}softmax(O_i) = \frac{e^{(O_i)}}{n + \sum_{j\in I^+} e^{(O_j)}}, \tag{7}$$

where n is a hyper-parameter to control the smoothness of the distribution, we set $n = 1$ in the experiments. The benefit of applying pruning in conjunction with the smooth selection function is that it mitigates abrupt changes in the attention distribution. Based on our definitions, we present the KL divergence between the original attention distribution and its pruned variants under two strategies. Let P denote the original softmax distribution over all tokens, defined as

$$P_i = \frac{e^{(O_i)}}{Z}, \quad \text{with } Z = \sum_{j=1}^{n} e^{(O_j)}. \tag{8}$$

After pruning, we retain a subset of tokens indexed by $I^+ \subset I = \{1, \ldots, n\}$, where $I^+ = I \setminus I^-$ and I^- denotes the set of pruned indices. The pruned attention is then re-normalized over the kept indices I^+, resulting in a sharper distribution:

$$Q_i = \frac{e^{(O_i)}}{Z_+}, \quad \text{where } Z_+ = \sum_{j\in I^+} e^{(O_j)}. \tag{9}$$

With our proposed n-softmax, the adjusted distribution becomes:

$$\tilde{Q}_i = \frac{e^{(O_i)}}{Z_+ + n}. \tag{10}$$

The KL divergence between the pruned attention distributions and the original distribution P can be explicitly characterized as:

$$\mathrm{KL}(Q \parallel P) = \log\left(\frac{Z}{Z_+}\right), \quad \mathrm{KL}(\tilde{Q} \parallel P) = \log\left(\frac{Z}{Z_+ + n}\right), \tag{11}$$

which yields the following expression for the difference in distributional shift:

$$\mathrm{KL}(Q \parallel P) - \mathrm{KL}(\tilde{Q} \parallel P) = \log\left(\frac{Z_+ + n}{Z_+}\right) > 0. \tag{12}$$

This inequality indicates that n-softmax more faithfully preserves the original attention distribution by compensating for the probability mass of pruned tokens. A complete derivation is provided in Appendix 6.

4 Experiments

4.1 Benchmark

We evaluate our method on MileBench [34] (MLLM), a benchmark specifically designed to assess long-context capabilities in multimodal language models. It collects widely-used datasets, providing a versatile and realistic foundation for evaluating model inference performance through two evaluation sets, *diagnostic* and *realistic-crafted*, which systematically measure inference performance in multi-modality scenarios. The tasks in the benchmark organized with:

- **Temporal Multi-Image Tasks (T1-T4)**: Temporal tasks involve understanding and predicting sequential events across images, and the methods in handling in action recognition, object tracking and spatial navigation.
- **Semantic Multi-Image Tasks (S1-S4)**: Semantic tasks focus on interpreting multimodal information, requiring inference methods to reason the knowledge-based QA, text-rich image analysis, visual relationship inference, dialogue understanding, and spatial reasoning.
- **Needle in a Haystack (NH)**: Retrieval tasks designed to find a preset password from a long context, which test kv cache inference methods in precise password extraction.
- **Image Retrieval (IR)**: Focused on identifying target images from candidates, which assess KV cache methods' effectiveness in perceptual and conceptual recognition.

Details of these challenging and comprehensive multimodal tasks, which include the corresponding datasets and evaluation metrics, are provided in the Appendix.

4.2 Baselines

We compare our KV cache pruning method with previous mainstream baselines include SnapKV [20], H2O [46], ReCo [31], and LOOK-M [39], each offering unique strategies for managing KV cache in long-context scenarios. SnapKV [20] introduces a method for intelligently identifying attention allocation patterns and compressing the KV cache by pooling essential tokens for extended sequences. H2O [46] presents a KV cache eviction policy that dynamically balances recent and frequently accessed tokens, identifying "heavy hitters" solely based on attention scores. ReCo [31] focuses on enhancing the efficacy of existing eviction policies through refined importance score calculations and carefully constructed eviction scopes, proposing a robust cache omission policy rooted in temporal attention scores and robustness measures. The LOOK-M family [39] considers modality awareness to evict image tokens and merge them with text tokens. This method prioritizes the retention of text-based KV pairs while evicting image-based KV pairs. The merging strategies include Average (A), Pivotal (P), and Weighted (W) merging.

4.3 Setup

We employed LLaVA-v1.5-7b [25] on RTX-4090 GPUs with *flash-attn-2.4.3post1* and LLaVA-v1.5-13b [25] on A100 GPUs with *flash-attn-2.6.3* [2] to conduct our experiments. To maintain consistency in generation, we set the sampling method to deterministic with a fixed temperature of 0, and the maximum context length was configured to 4096 tokens. Batch sizes were dynamically set based on dataset characteristics to balance computational load and memory constraints. Specifically, for datasets MMCoQA [21], NeedleInA-Haystack [34], and GPR1200 [32], the batch size was set to 1, while for all other datasets, a batch size of 24 was employed. Additionally, we calibrated the top-k value selection ratio for self-attention and cross-attention based on the sample mean ratio of cross-self region, with ablation studies showing the efficacy of adjusting this ratio. We applied biases of 0.5 and 1.5 in EgocentricNavigation [19] and SlideVQA [36], respectively, while keeping the default settings for other datasets. Our pre-processing and evaluation pipeline follows the standards of the benchmark, ensuring consistent assessment across the widely-used 29 datasets.

4.4 Main results

We use the widely-adopted open-source vision-language model LLaVA [25] to test KV cache performance on the benchmark. We present the results of our experiments in Table 1 and summarize our findings as follows.

Results Comparison. For the LLaVA-v1.5-7b model, our approach achieves notable improvements across several tasks. By independently selecting top-k tokens for cross-attention and self-attention, our method effectively retains key tokens specific to each modality. This separation enables the model to capture essential temporal sequences in tasks with sequential dependencies while simultaneously focusing on relevant multimodal content in tasks requiring semantic understanding. This result reveals that separating cross and self attentions allows for better retention of modality-specific cues, enhancing performance in tasks highly relevant to visual and textual data, with improvements of 4.5%, 7.2%, and 9.8% in T-3, S-5, and 4.5%, 7.2%, 9.8% improvement in NH task respectively. For the larger model, LLaVA-v1.5-13b, our approach shows even more pronounced improvements, especially on tasks T3, T4, and IR. These tasks share a common demand for precise handling of spatial and sequential elements across visual and textual modalities. By separating cross-attention and self-attention during top-k selection, our method effectively retains modality-specific tokens, which is crucial for tasks requiring spatial alignment and temporal tracking. This selective retention allows the model to preserve essential visual details for spatial localization (T3) with a performance boost

[2] https://github.com/Dao-AILab/flash-attention

Method	T-1	T-2	T-3	T-4	S-1	S-2	S-3	S-4	S-5	NH	IR
					LLaVA-7b						
Base	39.8	46.0	32.2	37.8	56.9	33.3	12.6	23.4	60.5	4.7	4.3
H2O [46]	39.5	46.6	31.8	38.5	55.0	33.0	13.0	23.0	60.0	1.4	3.7
SnapKV [20]	39.6	46.0	31.5	40.6	54.6	33.6	13.0	20.0	60.0	1.4	3.7
ReCo [31]	39.7	46.1	31.8	38.5	55.0	33.0	12.6	22.8	60.0	4.7	4.3
LOOK-M (A-Merge) [39]	39.7	46.1	32.2	39.1	54.9	34.0	12.4	21.4	60.5	1.6	3.7
LOOK-M (W-Merge) [39]	39.6	46.1	31.8	39.1	55.1	34.0	13.2	24.0	60.5	1.4	3.7
LOOK-M (P-Merge) [39]	39.7	46.1	32.5	39.9	57.0	34.0	12.8	23.9	60.5	5.3	3.8
LOOK-M (TP+A-Merge) [39]	39.7	46.1	32.0	39.0	56.5	33.8	12.9	23.1	60.5	5.1	3.5
LOOK-M (TP+W-Merge) [39]	39.8	46.1	32.5	39.9	57.0	34.0	12.8	23.9	60.5	5.3	3.8
LOOK-M (TP+P-Merge) [39]	39.8	46.1	32.5	39.9	57.0	34.0	12.8	23.9	60.5	5.3	3.8
CSP (Ours)	**39.9** (+0.1)	**46.8** (+0.2)	**32.5**	**41.6** (+1.0)	**57.5** (+0.5)	**34.1** (+0.1)	**13.7** (+0.5)	**27.8** (+3.8)	**61.0** (+0.5)	1.4	**6.3** (+2.0)
					LLaVA-13b						
Base	40.0	46.0	32.2	37.8	56.9	33.3	12.6	23.4	60.5	4.7	4.3
H2O [46]	39.5	45.9	30.4	47.9	64.1	48.7	13.9	25.1	59.7	3.6	0.0
SnapKV [20]	39.6	46.0	30.6	47.8	64.2	48.2	13.4	22.9	59.8	4.2	1.0
ReCo [31]	39.7	45.9	30.5	48.0	64.3	48.3	13.8	24.9	59.7	3.5	0.0
LOOK-M (A-Merge) [39]	39.7	46.1	30.7	48.0	64.6	48.0	13.3	22.1	59.8	4.6	1.0
LOOK-M (W-Merge) [39]	39.6	46.1	30.6	47.9	64.5	48.4	13.4	23.4	59.9	4.7	1.0
LOOK-M (P-Merge) [39]	39.7	46.0	30.6	48.0	64.6	48.0	13.3	25.7	59.8	5.8	1.0
LOOK-M (TP + A-Merge) [39]	39.7	46.2	30.7	48.0	65.4	48.3	13.7	26.6	60.0	11.2	1.0
LOOK-M (TP + W-Merge) [39]	39.8	46.1	30.8	48.1	64.8	48.2	13.9	26.9	60.0	11.4	1.0
LOOK-M (TP + P-Merge) [39]	39.8	46.2	30.8	48.1	65.2	48.5	14.1	26.6	60.0	11.7	1.0
CSP (Ours)	**40.0** (+0.2)	**46.5** (+0.3)	**32.4** (+1.6)	**49.0** (+0.9)	**65.4**	48.3	**14.4** (+0.3)	**27.0** (+0.1)	**60.5** (+0.5)	3.1	**9.6** (+8.6)

Table 1: Comparison of various KV cache management methods on the multiple multimodal tasks. Tasks are grouped as follows: Temporal Multi-Image Tasks (T1-T4), Semantic Multi-Image Tasks (S1-S4), Needle in a Haystack (NH), and Image Retrieval (IR). For fair comparison, we set the same KV cache size for all cache methods. Scores are calculated as the average performance across datasets within each subtask. For fair comparison, we set the same KV cache size for all inference methods.

of 8.3%, state transition (T4) with a 7.2% increase, and accurate retrieval in IR with a 8.6% improvement.

KV Cache Benefits. In long-context multi-modal benchmarks, it is often unnecessary to retain all key/value pairs in the KV cache throughout the decoding process. Recent works such as SnapKV [20], H2O [46], ReCo [31], and LOOK-M [39] have demonstrated that leveraging KV cache mechanisms can yield substantial performance gains. These results collectively indicate that a significant portion of visual tokens are redundant and contribute minimally to output quality. As experimantal results recorded in Table 1, strategically pruning such redundant tokens in the KV cache not only reduces computational overhead but also surpasses full-cache baselines, underscoring the importance of efficient cache design in vision-language models (VLMs) during the inference stage. The effect is particularly evident in the T-4 task, which focuses on counterfactual reasoning and tracking state changes, where all KV-cache-based methods consistently improve performance. Our approach achieves a notable +8.6% gain compare with other kv cache methods, establishing a new state-of-the-art for temporal multi-image understanding tasks by enhancing the model's ability to focus on critical object transitions and causal dynamics.

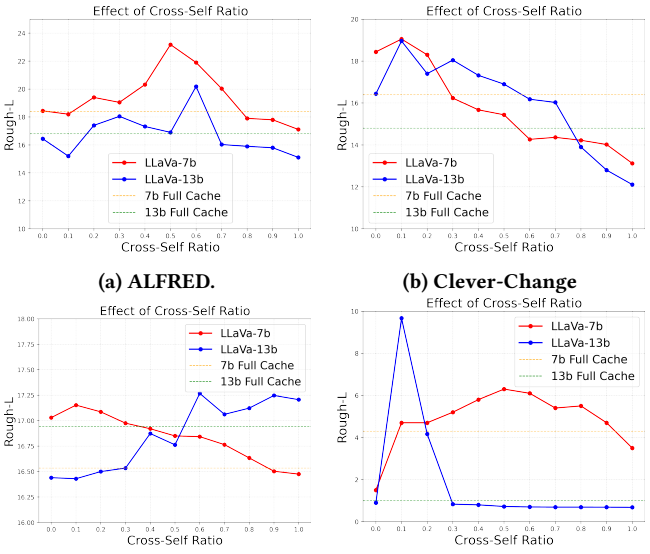
5 Ablation Study

We delve into ablation analysis of the KV cache approach to comprehensively assess our method. First, we present the hyper-parameters selection of K^c and K^s. Next, we assess speed latency and GPU memory usage to examine efficiency in a limited KV budget size. Futhermore, we test the pruning selection function, and record the impact of varying budget sizes on performance on different tasks. Finally, we present the influence of model architectures by introducing other open-source vision-language models.

5.1 Influence of the hyper-parameters

The remaining tokens in the cache are composed of the top-k indices selected independently from self-attention and cross-attention, which are then concatenated with recent tokens. Consequently, the hyperparameters of our method include the ratio of top-k selections between cross-attention and self-attention. We observe that performance reaches an optimal level when both cross-attention and self-attention tokens are selected in balanced proportions (i.e., the ratio is neither 0 nor 1) across 29 datasets. This configuration suggests that integrating both types of attention improves performance across datasets, as it allows the model to capture important tokens from multiple perspectives. However, there are differences between datasets; for instance, tasks focused on temporal alignment or sequential event detection (e.g. ALFRED), tend to benefit more from cross-attention, where context from multiple image frames is critical. On the other hand, datasets emphasizing individual object identification or attribute-focused reasoning are more reliant on self-attention, as they require maintaining a focused view on specific elements within a single frame.

In extreme cases (ratio = 1 for only self-attention K^s and ratio = 0 for only cross-attention K^c), we find distinct effects on task



(a) ALFRED. (b) Clever-Change

(c) Spot-the-Diff (d) GPR2000

Figure 4: The impact of the cross-self ratio.

Method	KV Budget	Decoding Latency	Cache Mem.
Full Cache	100%	26.023 ms/token	1.571 GiB
CSP	60%	24.377 ms/token	1.207 GiB
CSP	30%	21.027 ms/token	0.523 GiB
CSP	20%	19.736 ms/token	0.476 GiB
CSP	10%	16.287 ms/token	0.208 GiB

Table 2: We evaluate both latency and memory efficiency. Speed performance is tested using LLaVA-v1.5-7b on a single RTX 4090 with 100 warm-up iterations. The Cache Mem. refers to the additional memory used to store key/value pairs during the decoding stage.

performance. With self-attention, static tasks like visual recognition perform well due to precise frame-specific representations, though this setup struggles with tasks needing cross-frame integration. Conversely, cross-attention supports tasks requiring sequence alignment, like scene understanding, but can dilute focus on high-resolution details crucial for object-specific tasks.

5.2 Efficiency analysis

We analyze the efficiency of our proposed method in terms of decoding and GPU memory usage of the cache. Table 2 presents the results using LLaVA-v1.5-7b tested on a single RTX-4090 GPU with 100 warm-up iterations. The samples tested for the latency and memory usage is sampled from the long conversational embodied dialogue task [33] dataset in the benchmark. At a 60% budget, decoding latency is reduced to 24.377 ms/token, which provides a modest improvement over the 26.023 ms observed with a full cache, while memory usage drops by approximately 23% to 1.207GiB. As the cache budget decreases further, the benefits become more pronounced: at a 30% budget, latency reaches 21.027 ms per token and memory usage is nearly halved to 0.523 GiB. With the most aggressive reduction, using only 10% of the original cache, decoding

(a) n-Softmax on Top-k selection. (b) n-Softmax on CSP.

Figure 5: The benefit of n-softmax. We conduct the experiments on the ALFRED dataset by LLaVA-v1.5-7b.

latency improves to 16.287 ms per token, achieving a 37% speed increase over the full cache setup and an 87% reduction in memory usage to just 0.208 GiB. These findings illustrate that our method allows for a flexible balance between memory efficiency and decoding speed. Even with significantly reduced cache budgets, our approach retains acceptable latency and memory performance, offering a scalable solution for resource-constrained environments.

5.3 Influence of n-softmax

In this ablation study, we compare two KV cache pruning methods: one selects the top-k tokens directly from the overall selection function, while the other applies top-k selection separately within cross-attention and self-attention regions. We evaluate both approaches with standard Softmax and n-Softmax scoring functions to assess their impact on performance. Experimental results in Table 5 reveal that n-Softmax consistently provides a slight performance improvement over Softmax, indicating that the smooth transition positively impacts the pruning strategy during the decoding process. Specifically, selection-based approaches demonstrate clear benefits by focusing retention on high-value tokens, which enhances model efficiency. This effect is evident across tasks, as the separate top-k selection in cross and self-attention regions improves performance by capturing modality-specific important tokens more effectively. We find that this transition is particularly effective for tasks requiring both temporal coherence and fine-grained feature retention, as it allows for the selective pruning of large, less critical tokens under limited KV cache budgets. These results underscore the advantages of selection-based methods for KV cache pruning, especially when integrating cross-self separation with n-Softmax.

5.4 Impact of cache budget

As the cache budget increases, we observe consistent performance gains across all tasks, indicating that larger cache budgets enhance model accuracy and retrieval quality. For each dataset, performance improves steadily with an increase in cache size, moving closer to or exceeding the baseline set by full cache. In tasks with complex sequence dependencies like ALFRED, our method (CSP) achieves a significant boost in accuracy at higher cache budgets (60%), outperforming other methods and reaching a level above the full cache baseline. This pattern suggests that a larger cache budget is especially beneficial in scenarios where maintaining temporal coherence is crucial for task success. In tasks requiring fine-grained visual

differentiation, such as Spot-the-Diff and CLEVR-Change, performance gains with increased cache are less pronounced but still evident, indicating that these tasks can benefit from a moderate cache size. These findings support that our method consistently outperforms other methods across all budget size in the cache, suggesting that separate top-k selections from cross and self regions could effectively balance tokens selection and avoid collapse in the inference process.

5.5 Influence of Model Architectures

In this section, we evaluate the influence of model architecture on the performance of our proposed KV cache method across selected tasks (T-2, T-4, S-4, and IR) in the benchmark. We introduce two more architectures: InternVL-v1.5-7B [10], which scales up the vision encoder with cross-modal integration, and MobileVLM-V2-3B [12], which features a lighter structure with an efficient downsampling projector (LDPv2) and focuses on intra-modal processing. Table 3 demonstrates our test results of our methods compared with baselines. For InternVL-v1.5-7B, we observe that CSP achieves the highest performance in most tasks, particularly in T-2 (22.8) and S-4 (25.4), indicating that our KV cache pruning method benefits from InternVL's large-scale vision encoder. This architecture supports a robust cross-modal alignment, which CSP leverages by retaining crucial tokens independently from cross-attention and self-attention regions, maintaining contextual richness in visual-textual integration tasks. Regarding MobileVLM-V2-3B, CSP also demonstrates superior performance, especially in IR (5.3), where precise image retrieval benefits from MobileVLM's lightweight, modality-aware processing enabled by LDPv2. The lightweight design of this architecture allows CSP to perform well by focusing on high-salience tokens in the vision domain.

Method	T-2	T-4	S-4	IR
InternVL-v1.5-7B[10]				
Full Cache	19.2	21.3	19.1	0.0
H_2O [46]	20.0	20.4	19.6	0.5
SnapKV [20]	19.9	19.5	19.4	0.2
RoCo [31]	20.0	18.5	19.6	0.5
LOOK-M [31]	22.0	19.6	22.9	0.5
CSP (Ours)	**22.8**	**20.7**	**25.4**	**0.6**
MobileVLM-V2-3B [12]				
Full Cache	46.2	38.5	33.0	4.7
H_2O [46]	46.4	38.2	28.2	4.5
SnapKV [20]	46.4	38.5	27.2	4.7
RoCo [31]	46.6	38.0	28.9	4.6
LOOK-M [39]	47.0	38.7	32.8	4.8
CSP (Ours)	**47.3**	**39.0**	**33.1**	**5.3**

Table 3: Comparison of KV cache methods across a variety of tasks (T-2, T-4, S-4, IR) on InternVL-v1.5-7B [10] and MobileVLM-V2-3B [12].

6 Conclusion

In this work, we propose Cross-Self Pruning (CSP), a training-free KV cache method designed to independently optimize top-k tokens from cross-attention and self-attention regions. We employ the MileBench benchmark to evaluate our method, and the results demonstrate that CSP achieves competitive performance across all tasks while reducing cache budgets, which in turn largely improves the efficiency of multi-modal inference. Our ablation study further highlights the effectiveness of optimizing token selection through intra- and cross-modality pruning, validating the benefit of separating attention types and applying region-specific pruning strategies. Overall, CSP offers a simple yet effective design for improving the efficient-performance trade-off in vision-language model inference.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv:2303.08774* (2023).

[2] Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant Nair, Ilya Soloveychik, and Purushotham Kamath. 2024. Keyformer: Kv cache reduction through key tokens selection for efficient generative inference. *Proceedings of Machine Learning and Systems* 6 (2024), 114–127.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* 35 (2022), 23716–23736.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv:2309.16609* (2023).

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. *arXiv:2308.12966* (2023).

[6] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek LLM: Scaling open-source language models with longtermism. *arXiv:2401.02954* (2024).

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[8] Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, et al. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069* (2024).

[9] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16495–16504.

[10] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24185–24198.

[11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2023. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv:2312.14238* (2023).

[12] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766* (2024).

[13] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*. 5267–5275.

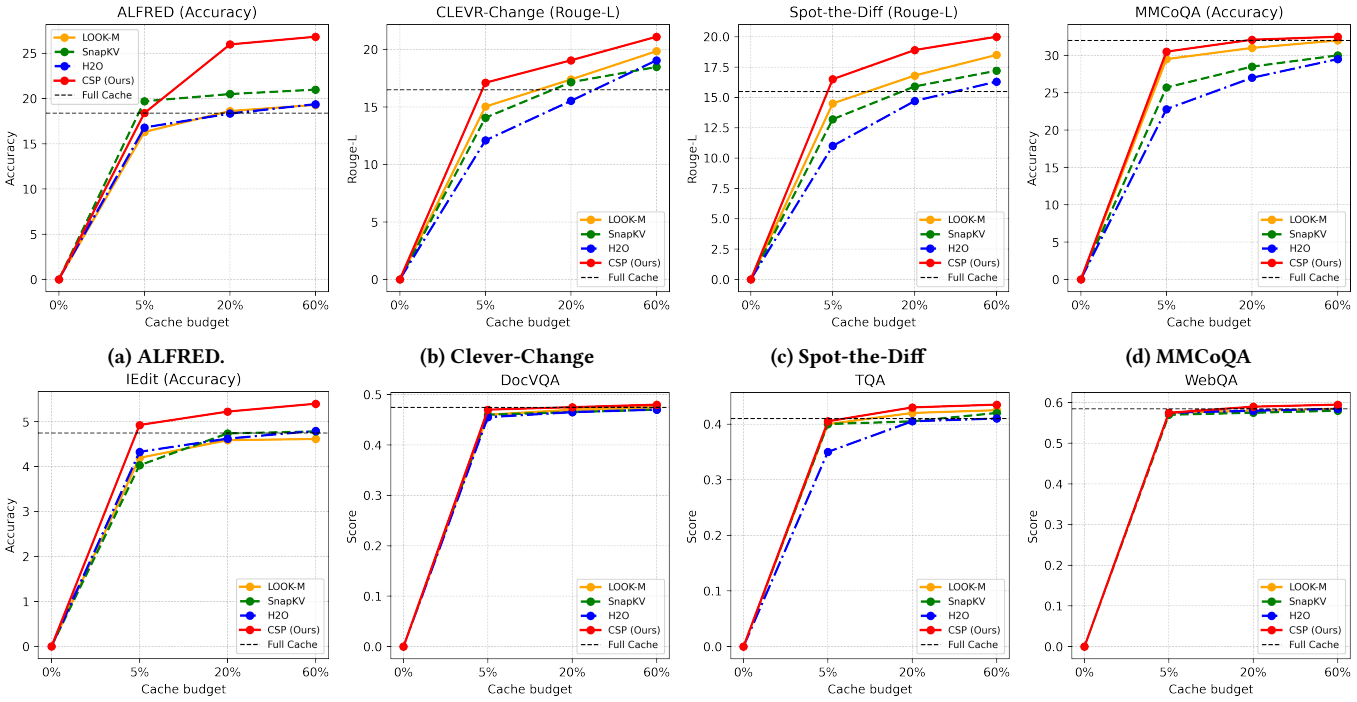
[14] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2023. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801* (2023).

[15] Mehrdad Hosseinzadeh and Yang Wang. 2021. Image change captioning by learning from an auxiliary task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2725–2734.

[16] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 709–727.

[17] Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. *arXiv preprint arXiv:1808.10584* (2018).

[18] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*. 4999–5007.



(a) ALFRED. (b) Clever-Change (c) Spot-the-Diff (d) MMCoQA

(e) IEdit (f) DocVQA (g) TQA (h) WebQA

Figure 6: The impact of the cache size budget.

[19] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 104–120.

[20] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469* (2024).

[21] Yongqi Li, Wenjie Li, and Liqiang Nie. 2022. Mmcoqa: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4220–4231.

[22] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814* (2024).

[23] Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. 2024. MiniCache: KV Cache Compression in Depth Dimension for Large Language Models. *arXiv preprint arXiv:2405.14366* (2024).

[24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved Baselines with Visual Instruction Tuning. *arXiv:2310.03744* (2023).

[25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26296–26306.

[26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems* 36 (2024).

[27] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2200–2209.

[28] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*. IEEE, 947–952.

[29] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. 2024. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems* 36 (2024).

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[31] Siyu Ren and Kenny Q Zhu. 2024. On the efficacy of eviction policy for key-value constrained generative language model inference. *arXiv preprint arXiv:2402.06262* (2024).

[32] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. 2022. GPR1200: a benchmark for general-purpose content-based image retrieval. In *International Conference on Multimedia Modeling*. Springer, 205–216.

[33] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

[34] Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. 2024. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532* (2024).

[35] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039* (2021).

[36] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 13636–13645.

[37] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805* (2023).

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[39] Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. *arXiv preprint arXiv:2406.18139* (2024).

[40] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 2024. Star: A benchmark for situated reasoning in real-world videos. *arXiv preprint arXiv:2405.09711* (2024).

[41] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305* (2023).

[42] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv:2304.14178* (2023).

[43] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. 2019. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442* (2019).

[44] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652* (2024).

[45] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. 2024. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. *arXiv preprint arXiv:2410.04417* (2024).

[46] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2024. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems* 36 (2024).

Appendix

Theoretical Justification of *n*-Softmax for Smoothing Pruned Attention

The pruning of low-scoring tokens from the key-value (KV) cache inherently modifies the attention distribution by reducing the normalization denominator in the softmax operation. This leads to a sharper distribution, where a few remaining tokens receive disproportionately high attention weights. To quantify this distortion, we analyze the Kullback–Leibler (KL) divergence between the pruned distributions and the original full softmax distribution. Let the original softmax over all tokens be defined as:

$$P_i = \frac{e^{O_i}}{\sum_{j=1}^{n} e^{O_j}} = \frac{e^{O_i}}{Z}, \tag{13}$$

where $Z = \sum_{j=1}^{n} e^{O_j}$ is the full partition function over all tokens. After pruning, we retain only a subset of indices $I^+ \subset \{1, 2, \ldots, n\}$, and the softmax is re-normalized over this subset. The pruned softmax becomes:

$$Q_i = \frac{e^{O_i}}{\sum_{j \in I^+} e^{O_j}} = \frac{e^{O_i}}{Z_+}, \quad \text{for } i \in I^+, \tag{14}$$

where $Z_+ = \sum_{j \in I^+} e^{O_j}$ is the new partition function over the retained tokens. Similarly, we define our proposed *n*-softmax as:

$$\tilde{Q}_i = \frac{e^{O_i}}{Z_+ + n}, \quad \text{for } i \in I^+, \tag{15}$$

where $n > 0$ is a smoothing constant that approximates the missing mass of the pruned tokens.

We now compute the KL divergence between the pruned distributions and the original full softmax P. For the standard softmax after pruning, we have:

$$\begin{aligned} \mathrm{KL}(Q \parallel P) &= \sum_{i \in I^+} Q_i \log\left(\frac{Q_i}{P_i}\right) \\ &= \sum_{i \in I^+} \frac{e^{O_i}}{Z_+} \log\left(\frac{e^{O_i}/Z_+}{e^{O_i}/Z}\right) \\ &= \sum_{i \in I^+} \frac{e^{O_i}}{Z_+} \log\left(\frac{Z}{Z_+}\right) \\ &= \log\left(\frac{Z}{Z_+}\right) \end{aligned} \tag{16}$$

Similarly, for the *n*-softmax case:

$$\begin{aligned} \mathrm{KL}(\tilde{Q} \parallel P) &= \sum_{i \in I^+} \tilde{Q}_i \log\left(\frac{\tilde{Q}_i}{P_i}\right) \\ &= \sum_{i \in I^+} \frac{e^{O_i}}{Z_+ + n} \log\left(\frac{e^{O_i}/(Z_+ + n)}{e^{O_i}/Z}\right) \\ &= \sum_{i \in I^+} \frac{e^{O_i}}{Z_+ + n} \log\left(\frac{Z}{Z_+ + n}\right) \\ &= \log\left(\frac{Z}{Z_+ + n}\right) \end{aligned} \tag{17}$$

Therefore, the difference between the two KL divergences is:

$$\begin{aligned} \mathrm{KL}(Q \parallel P) - \mathrm{KL}(\tilde{Q} \parallel P) &= \log\left(\frac{Z}{Z_+}\right) - \log\left(\frac{Z}{Z_+ + n}\right) \\ &= \log\left(\frac{Z_+ + n}{Z_+}\right) > 0 \end{aligned} \tag{18}$$

which holds for any $n > 0$. This result confirms that *n*-softmax yields a distribution that is strictly closer to the original full softmax than standard softmax after pruning. Intuitively, the additive constant n in the denominator serves to compensate for the missing contributions of pruned tokens, thereby preserving the smoothness and entropy of the attention distribution. This theoretical justification aligns with our empirical findings that *n*-softmax improves the stability and robustness of attention under aggressive token pruning.

A Ratio Selection for Dataset

The table reveals varying strategies for attention configuration across datasets, reflecting task-specific priorities. Datasets like *Spot-the-Diff* and *WebQA* emphasize cross-attention by assigning 90% of the top-k selection to cross-modal interactions. Conversely, tasks such as *ActionPrediction* rely entirely on intra-modal attention, with no top-k selection allocated to cross-attention. Overall, most datasets adopt a balanced approach, allocating 50% of the top-k selection to cross-attention and maintaining a recency bias of 1, indicating a general preference for equal weighting of intra- and cross-modal attention in multi-modal inference.

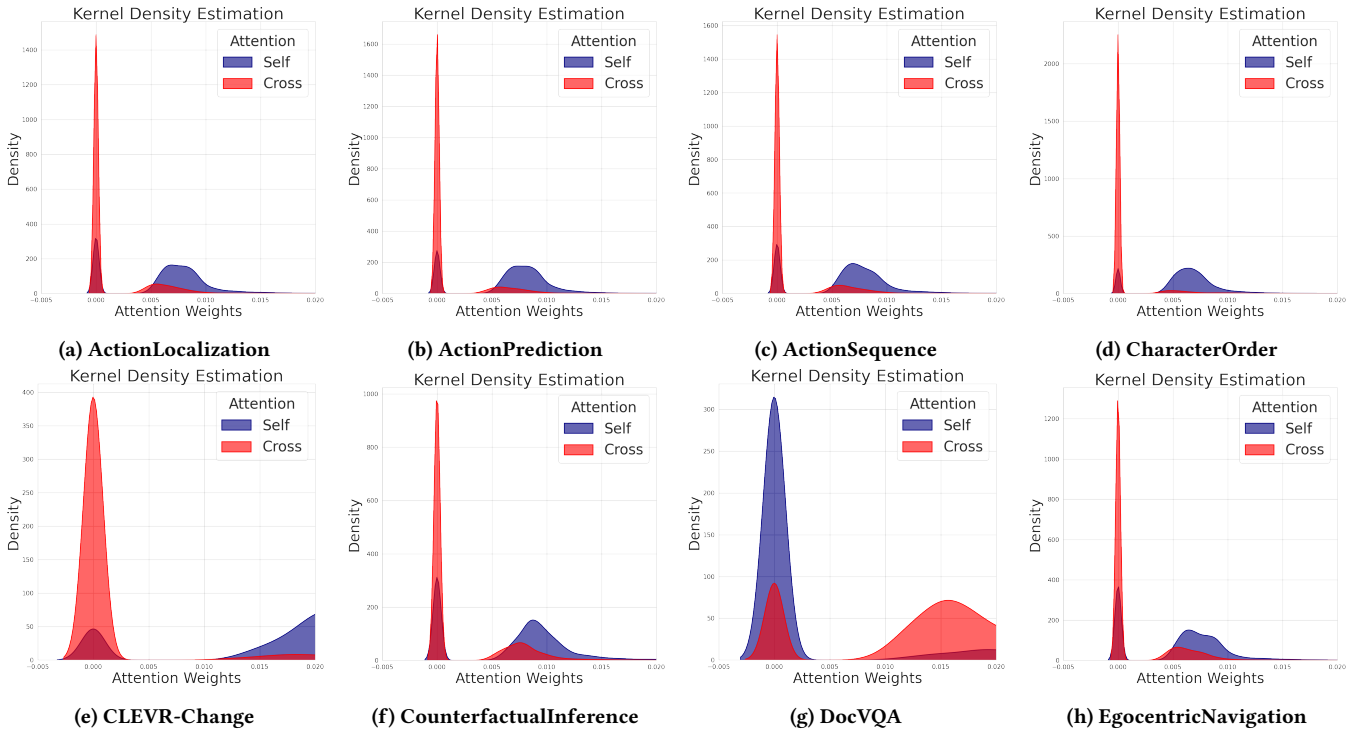
Model	T1	T2	T3	T4	S1	S2	S3	S4	NH	IR
LLaVA-7B	0.5	0.5	0.5	0.1	0.5	0.5	0.5	0.5	0.5	0.9
LLaVA-13B	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.9

Table 4: Cross Ratio Selection for Different Tasks.

Modality aware of the Distribution

In this section, we visualize the distribution differences between intra- and cross-attention using kernel density estimation (KDE). From the visualizations, we observe that certain datasets exhibit a stronger reliance on cross-attention, while others depend more heavily on self-attention. Here we apply the Kernel Density Estimation (KDE) and Jensen-Shannon (JS) divergence to analysis the difference of distributions.

In terms of KDE, figure 7 demonstrates the attention weight distributions for both self-attention (blue) and cross-attention (red)



(a) ActionLocalization (b) ActionPrediction (c) ActionSequence (d) CharacterOrder

(e) CLEVR-Change (f) CounterfactualInference (g) DocVQA (h) EgocentricNavigation

Figure 7: Kernel Density Estimation (KDE) of the attention weight distributions.

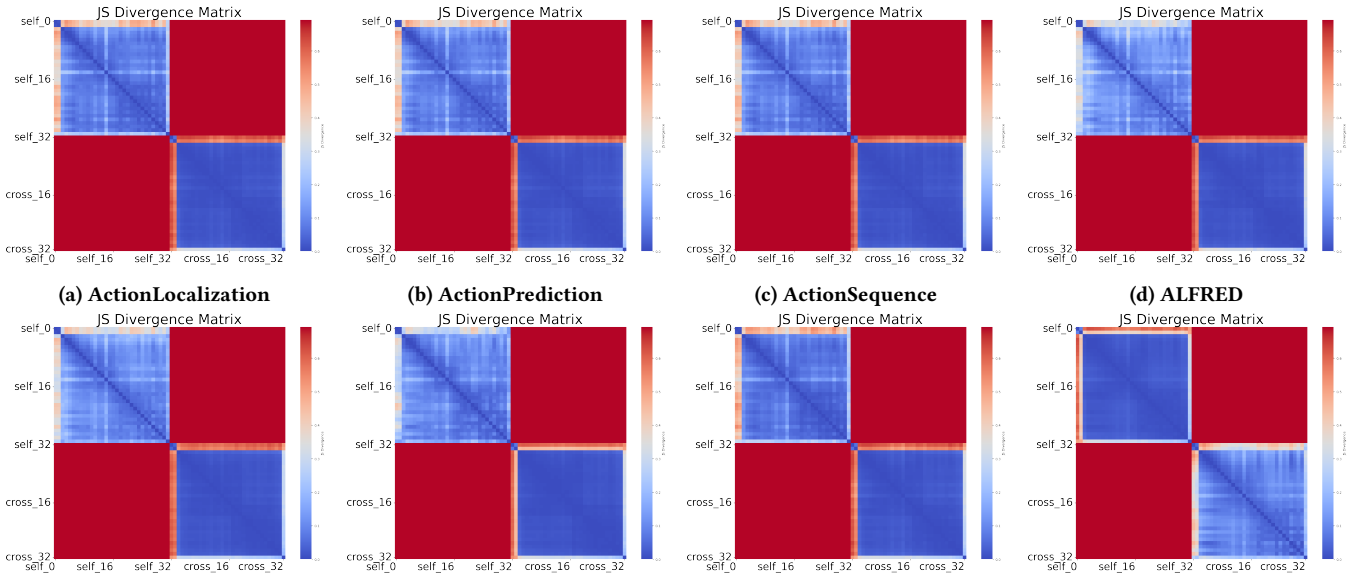
across eight datasets. It is evident that the two types of attention exhibit distinct patterns depending on the dataset, which directly impacts the pruning strategies during the KV cache process. For datasets such as *CLEVR-Change* and *CounterfactualInference*, cross-attention weights show a significantly concentrated and dominant peak at very low values, while self-attention demonstrates broader coverage. This suggests that cross-attention contributes heavily to the model's decision-making in these tasks, emphasizing token dependencies between modalities (e.g., image-text). Pruning strategies in these cases might inadvertently eliminate crucial cross-attention connections, leading to incomplete information transfer and subsequent degradation in inference accuracy. Conversely, datasets such as *DocVQA* and *EgocentricNavigation* reveal more dispersed and substantial self-attention weights, while cross-attention peaks remain narrow.

This indicates a reliance on intra-modal token interactions, such as contextual reasoning within the same modality. Aggressive pruning of self-attention tokens in such cases risks losing key intra-modal context, adversely impacting downstream predictions.

To further quantify the discrepancy between self-attention and cross-attention distributions, we compute the Jensen-Shannon (JS) divergence for each dataset. Higher divergence values suggest a stark imbalance between the two attention mechanisms, indicating that naive pruning may disproportionately affect one type of attention. In Figure 7, tasks like *CLEVR-Change* and *ActionPrediction* exhibit high divergence, implying the necessity for task-specific pruning thresholds to retain balanced contributions from both self- and cross-attention, whereas tasks like *DocVQA* show lower divergence, where self- and cross-attention operate more harmoniously, and uniform pruning strategies may suffice. These distribution discrepancies highlight several challenges and insights. Uniform pruning approaches that prioritize magnitude-based filtering may disproportionately affect regions with dense cross-attention peaks (e.g., *CLEVR-Change*), hindering the model's ability to encode cross-modal dependencies, particularly in visual reasoning tasks. For datasets where self-attention dominates (e.g., *EgocentricNavigation*), pruning strategies that overly prioritize low-weight tokens may reduce contextual coherence, especially for long-context sequences.

Furthermore, the differential pruning of self- and cross-attention tokens influences the effectiveness of KV cache utilization. Over-pruning either region may cause a skewed representation in the cache, reducing its utility for long-context inference.

The observed disparities in self- and cross-attention distributions across datasets necessitate an adaptive pruning framework that dynamically adjusts based on task-specific ratio, effectively balancing the preservation of essential intra- and cross-modal dependencies while optimizing KV cache utilization.



(a) ActionLocalization (b) ActionPrediction (c) ActionSequence (d) ALFRED

(e) CharacterOrder (f) CLEVR-Change (g) CounterfactualInference (h) DocVQA

Figure 8: Jensen-Shannon (JS) divergence scores between cross-attention and self-attention across all layers.

Table 5: Detailed Statistics and Taxonomy from MILEBench [34].

Category	Task	Dataset	Data Source	Count	Metric
Temporal Multi-image	Action Understanding and Prediction (T-1)	Action Localization Action Prediction Action Sequence	STA [13] STAR [40] STAR [40]	200	Accuracy
	Object and Scene Understanding (T-2)	Object Existence Object Interaction Moving Attribute Object Shuffle	CLEVRER [43] STAR [40] CLEVRER [43] Perception Test [29]	200	Accuracy
	Visual Navigation and Spatial Localization (T-3)	Egocentric Navigation Moving Direction	VLN-CE [19] CLEVRER [43]	200	Accuracy
	Counterfactual Reasoning and State Change (T-4)	Counterfactual Inference State Change Character Order Scene Transition	CLEVRER [43] Perception Test [29] Perception Test [29] MovieNet [16]	200	Accuracy
Semantic Multi-image	Knowledge Grounded QA (S-1)	Webpage QA Textbook QA Complex Multimodal QA Long Text with Images QA	WebQA [9] TQA [18] MultiModalQA [35] WikiVQA	200	Accuracy
	Text-Rich Images QA (S-2)	Slide QA OCR QA Document QA	SlideVQA [36] OCR-VQA [28] DocVQA [27]	200	Accuracy
	Visual Relation Inference (S-3)	Visual Change Captioning Visual Relationship Expressing	Spot-the-Diff [17] CLEVR-Change [15]	200	ROUGE-L
	Dialogue (S-4)	Multimodal Dialogue Conversational Embodied Dialogue	MMCoQA [21] ALFRED [33]	200	Accuracy
	Space Understanding (S-5)	nuScenes	nuScenes [7]	200	Accuracy
Diagnostic Evaluation	Needle In A Haystack (N-1)	Text Needle In A Haystack	TextNeedleInAHaystack	320	Accuracy
	Needle In A Haystack (N-2)	Image Needle In A Haystack	ImageNeedleInAHaystack	320	Accuracy
	Image Retrieval (I-1)	Image Retrieval	GPR1200 [32]	600	Accuracy