Reducing Peak Memory Usage for Modern Multimodal Large Language Model Pipelines

Anonymous ACL submission

Abstract

Multimodal large language models (MLLMs) achieve strong visual–textual reasoning by scaling to high-resolution images and long video sequences, but this scalability introduces substantial inference-time memory overhead due to the growth of the key–value (KV) cache. Existing KV-cache compression methods primarily operate after the full multimodal context has been processed, and therefore do not address the peak memory consumption incurred during the prefill stage. We observe that visual tokens in MLLMs exhibit strong structural regularities and representational redundancy that can be exploited earlier in the inference pipeline. Based on this observation, we propose a sequential, structure-aware KV-cache compression framework that operates during prefill and enforces a fixed memory budget throughout input processing. Experimental results show that our approach substantially reduces peak memory usage with minimal degradation in generative performance, enabling more practical and memory-efficient multimodal inference for large-scale visual inputs.

1 Introduction

Multimodal large language models (MLLMs) have emerged as a powerful paradigm for jointly reasoning over visual and textual inputs, enabling applications such as visual question answering (Antol et al., 2015), image-based reasoning (Shen et al., 2025), and video understanding (Zhang et al., 2024). To support these capabilities, modern MLLMs process increasingly complex visual signals, ranging from single images to high-resolution tiled patches and long video sequences. In a typical architecture (Liu et al., 2023), a pretrained vision encoder extracts visual features, an adaptor projects them into the language embedding space, and a transformer backbone jointly attends over vision tokens and textual inputs. While this unified attention enables flexible multimodal integration, it introduces substantial computational and memory challenges as the number of input tokens grows.

A key bottleneck arises from the self-attention operation (Vaswani et al., 2017), whose complexity scales quadratically with sequence length. Autoregressive transformers alleviate this cost through key–value (KV) caching (Pope et al., 2023), which stores intermediate attention representations and reduces per-token decoding complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. However, KV caching introduces a severe memory burden: the cache grows linearly with the number of tokens and must be retained across all layers and attention heads.

This challenge is particularly acute in multimodal settings. Recent advances in MLLMs have been driven by aggressively increasing the number of vision tokens, including tiled representations for high-resolution images (Bai et al., 2023; Chen et al., 2024; Tong et al., 2024), dense frame sampling for videos (Xu et al., 2024; Zhang et al., 2024; Yang et al., 2025b), and multi-view visual inputs (Cheng et al., 2025; Huang et al., 2025). These design choices substantially inflate the token count before decoding begins, causing the KV cache constructed during input processing to dominate memory usage. As a result, the prefill stage—where the full multimodal prefix is encoded—becomes the point of peak memory consumption during inference.

Prior work has sought to reduce inference-time memory usage primarily through KV-cache compression (Li et al., 2024; Kim et al., 2025a; Wan et al., 2025a). These methods exploit redundancy by evicting, merging, or approximating cached key–value pairs and are effective for long-context decoding. However, they typically apply compression only after the entire multimodal context has been processed, leaving the peak memory spike during prefill unaddressed. Token pruning methods (Yang et al., 2025a; Zhang et al., 2025a) reduce memory by discarding input tokens, but operate at the input level and ignore the heterogeneous roles that

different layers and attention heads assign to tokens (Yoon et al., 2025; Zhang et al., 2025b; Kaduri et al., 2025), increasing the risk of removing structurally important information.

In this work, we argue that the prefill stage itself offers untapped opportunities for memory-efficient multimodal inference. Visual inputs exhibit strong structural regularities: images consist of spatially coherent regions, and videos contain substantial temporal redundancy across frames. These structures form coarse-to-fine representations of the same underlying content, and not all visual tokens contribute equally to downstream reasoning.

Motivated by this observation, we propose a prefill-aware KV-cache compression framework that operates sequentially under a fixed memory budget. For single-turn settings, we introduce a query-aware strategy that leverages the textual prompt during prefill to estimate token importance and retain visually salient regions. For potential multi-turn interactions, where query signals may be unavailable, we further explore a query-agnostic variant that relies solely on the structural and representational properties of visual tokens. Together, these approaches substantially reduce peak memory usage during inference while preserving downstream performance, enabling scalable and memory-efficient multimodal inference across diverse interaction patterns.

2 Preliminaries

2.1 KV Cache in Transformer Inference

Transformers (Vaswani et al., 2017), as used in large language models (Brown et al., 2020), generate tokens autoregressively. At each step, self-attention computes interactions between the current query and all previously generated tokens. For a sequence of length t, attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad (1)$$

where Q, K, and V denote query, key, and value matrices, and d_k is the key dimension. During generation, only the query for the current token is newly computed, while keys and values from all preceding tokens are reused. To avoid recomputation, these keys and values are stored in GPU memory as a key–value (KV) cache.

2.2 The Necessity of KV Cache Management

KV caching reduces per-token decoding complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, but introduces a memory overhead that scales linearly with sequence length and model size. The total KV-cache memory footprint can be approximated as:

$$\begin{aligned}\text{Memory}_{\text{KV}} \approx 2 \times \text{layers} \times \text{heads} \times \text{dim}_{\text{head}} \\ \times \text{precision} \times \text{sequence length},\end{aligned} \quad (2)$$

where the factor of 2 accounts for both keys and values. As models scale to ultra-long contexts (e.g., 100K+ tokens), the KV cache alone can exceed available GPU memory, making effective KV-cache management essential for inference under fixed memory budgets.

2.3 The Vision Token Explosion Problem

Modern multimodal large language models (MLLMs) support high-resolution images and long video sequences through dense visual tokenization, resulting in substantially longer input sequences than text-only models. High-resolution images are decomposed into spatial grids of patches, each represented as a vision token. For an image of resolution $H \times W$, the number of vision tokens is:

$$N_{\text{vis}} = \frac{H \times W}{P^2}, \quad (3)$$

where P is the patch size. For example, an 4K image (3840×2160) yields over $42{,}000$ vision tokens with $P = 14$, demonstrating how visual inputs can dominate the token budget before decoding begins and drive peak memory usage during prefill.

3 Methodology

We propose a prefill-aware inference framework that reduces peak memory usage in multimodal large language models (MLLMs) by enforcing a fixed KV-cache budget throughout input processing, rather than compressing the cache only after the full multimodal context has been encoded.

3.1 Block-wise Processing for MLLMs

Conventional KV-cache eviction strategies construct the full KV cache before pruning, leading to high peak memory usage and frequent out-of-memory failures during prefill—particularly in MLLMs, where high-resolution images and long videos introduce thousands of vision tokens.

To address this issue, we adopt block-wise prefill (Kim et al., 2024, 2025b), partitioning the input sequence into contiguous blocks that are processed sequentially, as summarized in Alg. 1. After each block is encoded, its KV pairs are appended to the

Algorithm 1 Block-wise Prefill with KV Eviction

Input: Input sequence S, Block size b, Memory budget M
Output: Compressed KV Cache $\mathcal{C}$
Partition S into blocks $\{B_1, B_2, \ldots, B_N\}$ of size b
$\mathcal{C} \leftarrow \emptyset$ ▷ Initialize empty cache
for each block $B_i \in \{B_1, \ldots, B_N\}$ **do**
  $(K_i, V_i) \leftarrow \text{ComputeKV}(B_i)$ ▷ Generate KV pairs for current block
  $\mathcal{C} \leftarrow \mathcal{C} \cup (K_i, V_i)$ ▷ Append new pairs to cache
  if $|\mathcal{C}| > M$ **then**
    $k_{excess} \leftarrow |\mathcal{C}| - M$
    $\mathcal{C} \leftarrow \text{Evict}(\mathcal{C}, k_{excess})$ ▷ Reduce cache to budget M
  end if
end for
return $\mathcal{C}$

cache and pruned to satisfy a fixed budget M. This explicitly bounds the KV-cache size throughout prefill, preventing peak memory growth.

Block-wise prefill is well suited to multimodal inputs. Unlike text, visual inputs exhibit strong structural organization: images consist of spatially coherent tiles, and videos of temporally contiguous frame groups. We align block boundaries with these visual structures, enabling eviction decisions to be made at semantically meaningful granularity and improving robustness to compression.

3.2 Eviction Strategies

Within the block-wise framework, we consider two complementary eviction strategies that differ in their reliance on query information. Both operate online during prefill and are applied immediately after each block.

Query-Aware Eviction. For single-turn settings, we adopt a query-aware eviction strategy based on SnapKV (Li et al., 2024). Proxy query tokens are extracted from the textual prompt and used to compute cross-attention over cached keys. Given query features q_{obs} and cached key k_j, the importance score is:

$$\alpha_j = \text{Softmax}\left(\frac{q_{\text{obs}} \cdot k_j^\top}{\sqrt{d_k}}\right). \quad (4)$$

Tokens with lower importance scores are evicted until the cache satisfies the budget M. Applied sequentially during prefill, this strategy prioritizes visually salient regions relevant to the task while discarding redundant tokens early.

Method (KV Budget)	ImageNeedle	V*	MLVU	Video-MME (L)	Average	Δ
InternVL3.5-8B						
Full Cache	80.31	84.35	51.28	53.89	67.46	-
SnapKV (4096)	80.94	82.72	50.00	52.33	66.50	0.96
SnapKV (2048)	80.31	**83.76**	49.61	52.33	66.50	0.96
SnapKV (1024)	80.00	82.61	**51.00**	**53.11**	**66.68**	**0.78**
KeyDiff (4096)	**83.13**	74.87	49.60	51.33	64.03	3.43
KeyDiff (2048)	79.69	75.39	50.40	52.00	65.23	2.23
KeyDiff (1024)	74.06	74.35	50.40	52.22	62.76	4.70
Qwen2.5-VL-7B						
Full Cache	83.70	79.58	48.80	50.00	65.52	-
SnapKV (4096)	**85.00**	78.53	44.82	48.77	64.28	1.24
SnapKV (2048)	72.19	78.53	45.42	49.11	61.31	4.21
SnapKV (1024)	45.31	76.96	46.02	**49.56**	54.46	11.06
KeyDiff (4096)	81.56	**79.58**	**47.41**	49.33	**64.47**	**1.05**
KeyDiff (2048)	78.44	69.63	46.41	48.00	60.62	4.9
KeyDiff (1024)	66.25	67.02	43.43	46.55	55.82	9.7

Table 1: **Performance under fixed KV-cache budgets.** Best results are in bold. Δ denotes the difference from the full-cache baseline. Our method maintains stable performance across compression settings, with minimal degradation even at a budget of 1024 ($\sim 90\%$ compress)

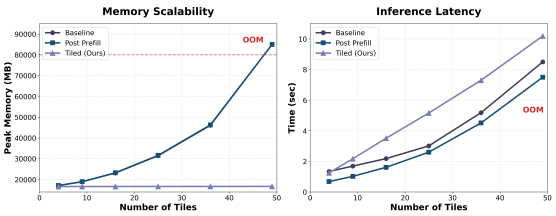

Figure 1: **Peak memory usage and inference latency** as the number of image tiles increases (InternVL-3.5). Our method maintains nearly constant peak memory during prefill under a fixed KV-cache budget, preventing out-of-memory (OOM), at the cost of increased inference latency due to sequential processing.

Query-Agnostic Eviction. For potential multi-turn scenarios where query signals may be unavailable, we employ a query-agnostic strategy based on KeyDiff (Park et al., 2025). The method preserves representational diversity by retaining keys that deviate most from the average representation. Specifically, we define an anchor vector μ as the mean of cached keys and prioritize retention of keys with lower similarity to μ. This avoids $\mathcal{O}(N^2)$ pairwise comparisons while preserving outliers and rare visual features without relying on query information.

4 Experiments

4.1 Benchmark Performance

Benchmarks. We evaluate on benchmarks that are sensitive to the scale and structure of visual tokens. For images, we use ImageNeedleInHaystack from MileBench (Song et al., 2024) and V* (Wu

(a) Forward under budget

Method (KV Budget)	ImageNeedle
Block Forward (1024)	80.94
Bulk Forward (1024)	80.31

(b) Static vs. Dynamic

Method (KV Budget)	ImageNeedle
Static (1024)	80.31
Dynamic (1024)	74.68

(c) Input res. vs. Compression

Method (KV Budget)	ImageNeedle
Compression (1024)	80.31
Reduction (1024)	9.38

Table 2: **Analysis of prefill strategies under a fixed KV-cache budget.** (a) Forward execution strategies, (b) static vs. dynamic budgeting, and (c) input resolution reduction vs. prefill-stage compression.

Block Size (KV Budget)	ImageNeedle	Global Peak (GB)	Avg. Peak (GB)
Qwen2.5-VL-7B			
256 (2048)	72.19	17.80	17.12
512 (2048)	75.31	18.00	17.19
784 (2048)	80.63	18.21	17.26
1024 (2048)	79.38	18.38	17.37

Table 3: **Effect of block size** under a fixed KV-cache budget (Qwen2.5-VL-7B). Performance peaks at block size 784, which matches the model's native visual tokenization.

and Xie, 2024), which require dense visual localization. For videos, we adopt MLVU (Zhou et al., 2024) and the long-video setting of Video-MME (Fu et al., 2025). All experiments are conducted on NVIDIA A100 GPUs using standard evaluation protocols. We report task accuracy, average accuracy, and the difference Δ relative to the full-cache baseline.

Main results. We evaluate InternVL3.5-8B (Wang et al., 2025) and Qwen2.5-VL-7B (Bai et al., 2025). InternVL3.5-8B is tested with up to 36 image tiles (9,216 vision tokens) and 32 video frames (8,192 tokens), while Qwen2.5-VL-7B uses up to 8,192 vision tokens for both modalities. Unless stated otherwise, the block size is 256.

As shown in Tab. 1, our prefill-stage compression preserves performance under aggressive KV-cache budgets, achieving up to $\sim$90% compression. Fig. 1 shows that peak KV-cache memory remains nearly constant as image tiles increase, whereas baseline methods grow linearly and encounter out-of-memory failures beyond 36 tiles. These results demonstrate that our method controls peak memory usage during prefill without sacrificing accuracy.

4.2 Result Analysis

Query-aware vs. query-agnostic eviction. Query-aware eviction (SnapKV) achieves the strongest performance when query signals are available, particularly at small budgets. On InternVL3.5-8B, SnapKV at budget 1024 incurs only a 0.78 average accuracy drop. The query-agnostic KeyDiff variant remains competitive, with modest degradation at larger budgets (e.g., 3.43 at 4096), indicating that preserving representational diversity alone retains task-relevant information and supports multi-turn settings.

Video tasks and non-monotonic behavior. On video benchmarks, reducing the cache budget does not always degrade performance monotonically. For InternVL3.5-8B, SnapKV at budget 1024 achieves slightly improved results on MLVU and Video-MME, suggesting that prefill-stage compression suppresses redundant temporal information and yields more focused representations.

Latency and budgeting. Block-wise prefill increases latency due to sequential execution; however, a hybrid strategy that processes the first M tokens in a single forward pass (budget 1024) yields comparable accuracy (80.94 vs. 80.31 on ImageNeedle; Tab. 2(a)) and is used by default. Dynamic layer-wise budgeting (Li et al., 2025; Wan et al., 2025b) underperforms static allocation during prefill (5.63-point drop at budget 1024; Tab. 2(b)), likely due to incomplete attention statistics.

Compression vs. Input reduction. Reducing input resolution under the same KV-cache budget causes severe performance degradation (9.38 accuracy on ImageNeedle; Tab. 2(c)), whereas prefill-stage compression preserves high-resolution visual information while controlling memory usage.

Block size and structural alignment. Block size has a strong impact on compression effectiveness. For Qwen2.5-VL-7B, accuracy peaks at block size 784 under a budget of 2048, which exactly matches the model's native 28×28 visual tokenization. In contrast, block sizes that are misaligned with this tokenization (e.g., 512) lead to reduced robustness, explaining the larger performance drop observed for Qwen2.5-VL-7B in Tab. 1. This result highlights that compression granularity must align with the spatial structure of visual representations, and that vision-aware block design is critical for maintaining performance.

5 Conclusion

We propose a prefill-aware, block-wise KV-cache compression method achieving up to $\sim$90% cache reduction with minimal performance loss.

Limitations

Our method enforces a fixed KV-cache budget during prefill and therefore introduces several natural trade-offs. Block-wise prefill processes inputs sequentially, which can increase inference latency compared to bulk execution, reflecting an inherent memory–latency trade-off; in practice, this overhead can be mitigated with hybrid execution strategies. In addition, compression effectiveness depends on alignment between block boundaries and the structure of visual representations, and query-agnostic eviction prioritizes general representational diversity rather than task-specific relevance. Finally, our approach focuses on inference-time optimization without modifying training, and models explicitly trained with prefill-stage compression may further improve robustness.

Ethical Considerations

This work focuses on improving inference-time memory efficiency for multimodal large language models through KV-cache management. The proposed method does not introduce new model capabilities, training data, or deployment scenarios, and does not alter model behavior beyond resource usage. As such, it does not raise additional ethical concerns beyond those already associated with large language models and multimodal systems in general.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 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*, pages 24185–24198.

An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, and 1 others. 2025. 3d aware region prompted vision language model. *arXiv preprint arXiv:2509.13317*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Jiaxin Huang, Runnan Chen, Ziwen Li, Zhengqing Gao, Xiao He, Yandong Guo, Mingming Gong, and Tongliang Liu. 2025. Mllm-for3d: Adapting multimodal large language model for 3d reasoning segmentation. *arXiv preprint arXiv:2503.18135*.

Omri Kaduri, Shai Bagon, and Tali Dekel. 2025. What's in the image? a deep-dive into the vision of vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14549–14558.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. 2025a. Kvzip: Query-agnostic kv cache compression with context reconstruction. *arXiv preprint arXiv:2505.23416*.

Minsoo Kim, Arnav Kundu, Han-Byul Kim, Richa Dixit, and Minsik Cho. 2025b. Epicache: Episodic kv cache management for long conversational question answering. *arXiv preprint arXiv:2509.17396*.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. 2024. Infinipot: Infinite context processing on memory-constrained llms. *arXiv preprint arXiv:2410.01518*.

Kunxi Li, Yufan Xiong, Zhonghua Jiang, Yiyun Zhou, Zhaode Wang, Chengfei Lv, and Shengyu Zhang. 2025. Flowmm: Cross-modal information flow guided kv cache merging for efficient multimodal context inference. *Preprint*, arXiv:2511.05534.

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. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in*

neural information processing systems, 36:34892–34916.

Junyoung Park, Dalton Jones, Matthew J Morse, Raghavv Goel, Mingu Lee, and Chris Lott. 2025. Keydiff: Key similarity-based kv cache eviction for long-context llm inference in resource-constrained environments. *Preprint*, arXiv:2504.15364.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model, 2025. *URL https://arxiv.org/abs/2504.07615*.

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*.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, and 1 others. 2024. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. 2025a. Meda: Dynamic kv cache allocation for efficient multimodal long-context inference. *arXiv preprint arXiv:2502.17599*.

Zhongwei Wan, Hui Shen, Xin Wang, Che Liu, Zheda Mai, and Mi Zhang. 2025b. Meda: Dynamic kv cache allocation for efficient multimodal long-context inference. *Preprint*, arXiv:2502.17599.

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. *Preprint*, arXiv:2406.18139.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Penghao Wu and Saining Xie. 2024. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv*.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025a. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, and 1 others. 2025b. Cambrian-s: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*.

Heeji Yoon, Jaewoo Jung, Junwan Kim, Hyungyu Choi, Heeseong Shin, Sangbeom Lim, Honggyu An, Chaehyun Kim, Jisang Han, Donghyun Kim, and 1 others. 2025. Visual representation alignment for multimodal large language models. *arXiv preprint arXiv:2509.07979*.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2025a. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20857–20867.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. *Preprint*, arXiv:2306.14048.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. 2025b. Cross-modal information flow in multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19781–19791.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv e-prints*, pages arXiv–2406.

Appendix

A Related Works

The rapid growth of the key–value (KV) cache in long-context inference has motivated extensive research on cache compression methods (Li et al., 2024; Zhang et al., 2023; Park et al., 2025; Xiao et al., 2023), commonly categorized into quantization-, eviction-, and merging-based approaches. This work focuses on eviction-based methods, with emphasis on their limitations in multimodal large language models (MLLMs).

KV-Cache Eviction in LLMs. KV-cache eviction strategies maintain a fixed memory budget by selectively discarding tokens that are unlikely to contribute to future generation. Early methods such as StreamingLLM (Xiao et al., 2023) identify *attention sinks* and retain them together with a sliding window of recent tokens. More advanced approaches, including H_2O (Zhang et al., 2023) and SnapKV (Li et al., 2024), leverage accumulated attention statistics to preserve *heavy hitter* tokens that are frequently attended to during decoding.

Complementary query-agnostic strategies avoid reliance on a specific query signal. KeyDiff (Park et al., 2025) observes that highly attended tokens tend to be representationally diverse, and therefore retains keys that are distant from the centroid of the key distribution. Unlike query-dependent methods, such approaches enable the compressed KV cache to be reused across different queries.

KV-Cache Eviction in MLLMs. In multimodal settings, KV-cache eviction must additionally address the substantial redundancy introduced by large numbers of visual tokens. LOOK-M (Wan et al., 2024) exploits the tendency of MLLMs to prioritize textual tokens, selectively pruning visual tokens while preserving the text prompt. MEDA (Wan et al., 2025b) introduces layer-wise adaptive budget allocation guided by cross-modal attention entropy, allowing visually sensitive layers to retain denser representations. FlowMM (Li et al., 2025) further extends this direction by dynamically merging tokens based on cross-modal attention patterns. Together, these methods move beyond coarse window-based pruning toward more modality-aware KV-cache management, but primarily operate after the full multimodal context has been processed.

B Discussion

The results demonstrate that controlling peak memory during the prefill stage is both feasible and critical for scaling multimodal inference to high-resolution and long-context visual inputs. By shifting KV-cache compression from a post-prefill operation to an online, structure-aligned process, our framework enables models to retain high-resolution visual information while operating under strict memory budgets. The consistent performance observed across image and video benchmarks, together with stable peak memory usage and the avoidance of out-of-memory failures, suggests that prefill-aware compression addresses a fundamentally different bottleneck than existing decoding-time methods. More broadly, these findings indicate that memory efficiency in MLLMs is not solely a function of final cache size, but is strongly shaped by how and when visual context is processed during inference.