VISPARSE: RETRIEVAL-BASED VISUAL ATTENTION SPARSIFICATION FOR MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Extended reasoning enables multimodal large language models to solve complex visual tasks, but caching and repeatedly attending to thousands of visual tokens during long reasoning traces incur substantial memory and computational overhead. Token compression offers a promising way to reduce this overhead. However, existing methods can be limited by restricted selection flexibility and insufficient adaptation to visual attention that varies across reasoning steps. We introduce ViSparse, a retrieval-based framework for visual-attention sparsification. ViSparse combines head-specific hierarchical cluster retrieval with coverage-adaptive refresh to adapt to attention patterns during reasoning, selectively loading relevant visual token states from CPU to GPU. It further exploits attention similarity across adjacent layers to share retrieval indices, reducing redundant retrieval while allowing each layer to compute attention using its own visual token states. Across three multimodal reasoning models and eight benchmarks, ViSparse achieves the highest average accuracy among the evaluated compressed methods for each model, while maintaining accuracy comparable to full-cache attention. Efficiency experiments show up to $2.262\times$ decoding speedup and up to 70.1% lower peak GPU key–value cache storage. These results show that ViSparse improves multimodal reasoning efficiency while retaining on-demand access to all visual token states.

1 INTRODUCTION

Reasoning capabilities enable multimodal large language models (MLLMs), such as Gemini Comanici et al. (2025) and OpenAI's o-series Jaech et al. (2024), to more effectively solve complex visual tasks, such as chart question answering, spatial reasoning, and visual mathematical reasoning Jia et al. (2025); Zhao et al. (2026). However, generating extended reasoning traces requires additional autoregressive decoding steps, increasing both computational cost and inference latency. In multimodal settings, this overhead is compounded by high-resolution images and multi-image or video inputs, which can introduce thousands of visual tokens Zhang et al. (2026a); Shao et al. (2025). With full attention over the visual context, each decoding step attends to all visual tokens by reading their cached key–value (KV) states. Over long reasoning traces, the same visual states are accessed repeatedly, increasing the cumulative attention computation. At the same time, storing the visual KV cache for a high-resolution or multi-image input can occupy substantial GPU memory throughout generation. These computational and memory costs make visual-token efficiency particularly important for multimodal reasoning.

Token compression has emerged as a way to reduce the inference cost of multimodal large language models Shao et al. (2025). Existing approaches include merging similar representations or retaining representative tokens Wang et al. (2025a); Yang et al. (2025); Alvar et al. (2025), as well as selecting visual tokens based on attention-derived relevance scores Liu et al. (2024); Zhang et al. (2024); He et al. (2024). However, these static token compression methods struggle to accommodate the changing visual evidence requirements of long-chain reasoning Xu et al. (2026). Recent methods such as VisionPulse Xu et al. (2026) and HybridKV Zeng et al. (2026) incorporate dynamic visual-token selection during decoding. However, VisionPulse relies on early-layer attention signals and retains the full visual KV cache on the GPU. HybridKV uses constrained chunk-wise retrieval and, for efficiency, dynamically updates only part of the visual KV cache while statically compressing the rest

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

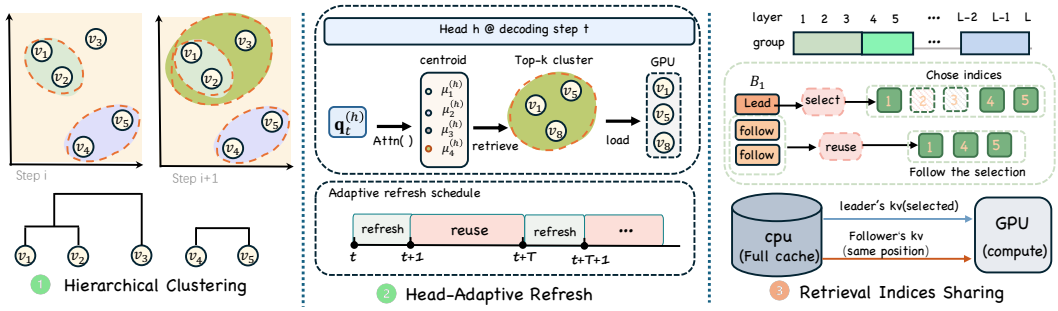


Figure 1: Overview of ViSparse. Left: head-specific hierarchical clustering organizes visual tokens for retrieval. Middle: query-driven cluster retrieval loads relevant visual tokens onto the GPU, with a head-adaptive schedule controlling refresh and reuse. Right: cross-layer sharing reuses the leader layer's selected token positions, while each layer loads its own KV states from CPU memory.

during prefill. These limitations leave room for improvement in multimodal reasoning performance, motivating methods that better account for evolving attention patterns during reasoning.

To identify the properties that an effective retrieval scheme should satisfy, we analyze visual attention across heads, decoding steps, and layers (Section 2). First, visual attention is both head-dependent and step-dependent: different heads emphasize different regions, and a region that receives little attention at one step may become important at a later step. These observations suggest that effective sparsification for multimodal reasoning should enable fine-grained, dynamic token selection from the full visual context for every attention head. Second, adjacent layers exhibit highly similar attention patterns. This similarity enables cross-layer reuse of token selections and motivates system optimizations tailored to this cross-layer structure.

Based on these observations, we introduce **ViSparse**, a retrieval-based visual-attention sparsification framework for multimodal reasoning (Fig. 1). ViSparse keeps the complete visual KV cache in CPU memory, preserving all visual tokens for dynamic selection by every head while keeping only a small subset active on the GPU. Specifically, head-specific hierarchical cluster retrieval Chen et al. (2024a) supports fine-grained selection from the full visual context, while coverage-adaptive refresh adjusts update frequencies to each head's attention coverage. Index sharing exploits cross-layer attention similarity to reuse token selections across adjacent layers. We further accelerate cluster selection and token gathering with fused GPU kernels, reduce redundant transfers through GPU buffer reuse, and overlap data movement with computation through asynchronous cross-layer prefetching.

We evaluate ViSparse on three multimodal reasoning models from the Qwen3-VL-Thinking and InternVL-3.5-Thinking families across eight benchmarks. ViSparse achieves the highest average accuracy among the evaluated compressed methods on all three models, outperforming the strongest compressed baseline for each model by 2.00–2.65 percentage points. On Qwen3-VL-Thinking-8B, it achieves up to 2.262× decoding speedup and reduces peak GPU KV-cache storage by 70.1% compared with FullKV. These results demonstrate that ViSparse substantially reduces the inference cost of multimodal reasoning while maintaining accuracy comparable to full-cache attention.

Our contributions are summarized as follows:

- We characterize head- and step-dependent visual-attention variation during reasoning and cross-layer similarity, motivating ViSparse.
- We propose **ViSparse**, which combines head-specific hierarchical cluster retrieval, coverage-adaptive refresh, and cross-layer index sharing to enable dynamic access to the full visual context for every head. A CPU-resident visual KV cache and tailored system optimizations reduce GPU memory and data-movement overhead.
- Across three models and eight benchmarks, ViSparse achieves the highest average accuracy among the evaluated compressed methods with a 10% visual-token budget. On Qwen3-VL-Thinking-8B, it delivers up to 2.262× decoding speedup and reduces peak GPU KV-cache storage by 70.1% compared with FullKV.

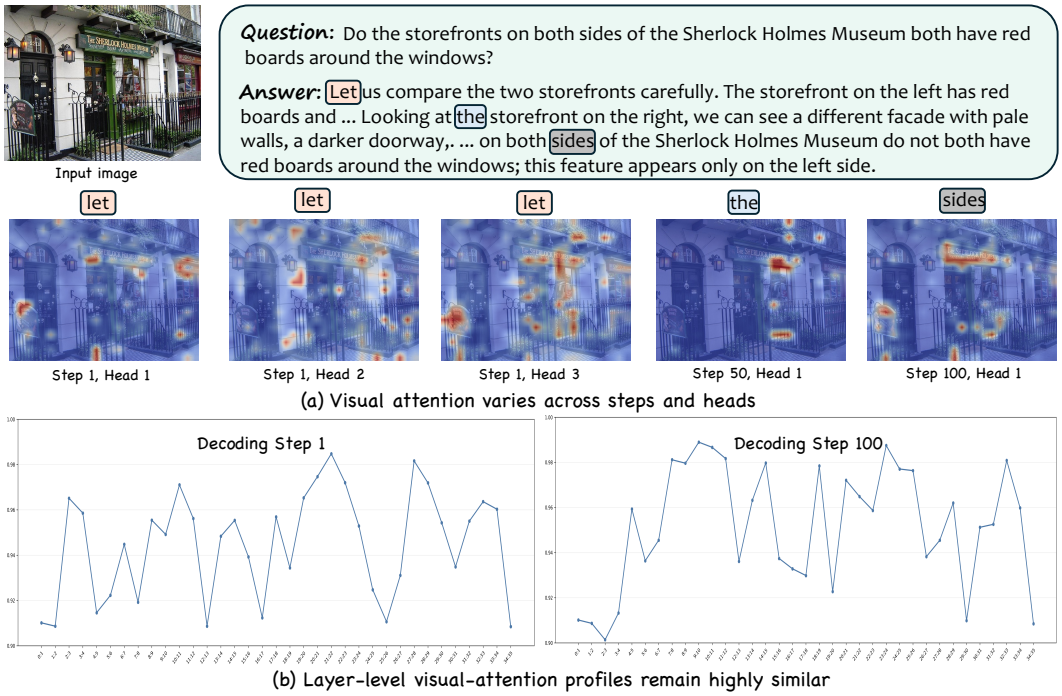


Figure 2: Motivation for ViSparse. (a) Visual attention is both head- and step-dependent: different heads focus on different image regions, and a head's focus changes across decoding steps. (b) Cosine similarities between adjacent layers remain high at decoding steps 1 and 100, revealing cross-layer redundancy that supports shared token selection.

2 Motivation

2.1 Visual Attention Varies Across Heads and Reasoning Steps

Our analysis on MileBench Song et al. (2024) reveals that visual attention varies across heads and reasoning steps. Figure 2(a) shows attention maps from the first layer: Heads 1–3 exhibit different spatial attention patterns at step 1, while Head 1 shifts its focus across steps 1, 50, and 100.

These observations highlight a key characteristic of visual attention during multimodal reasoning: each head focuses on a limited subset of visual tokens at a given step, but this subset varies substantially across heads and evolves over the reasoning trajectory. Static compression or dynamic updates restricted to only part of the visual KV cache may therefore fail to accommodate these evolving attention patterns.

2.2 Visual Attention Is Highly Similar Across Adjacent Layers

We further examine the similarity of visual attention across layers using Qwen3-VL-Thinking-8B on MileBench. For each input, we collect text-to-visual attention at every layer and compute the cosine similarity between the visual-attention vectors of adjacent layers. We report the mean similarity across inputs in the dataset.

As shown in Fig. 2, all adjacent layer pairs exhibit high cosine similarity, indicating similar visual-attention patterns across adjacent layers. The observation concerns layer-level attention distributions and is compatible with the head-specific variation discussed in Section 2.1.

Taken together, these observations reveal complementary properties of visual attention: its focus varies across heads and reasoning steps, while its patterns remain similar across adjacent layers. This cross-layer consistency suggests potential redundancy in independently identifying relevant

visual tokens at every layer, highlighting an opportunity to reduce selection overhead while accommodating the variation across heads and reasoning steps.

3 METHODOLOGY

3.1 OVERVIEW

Motivated by the observations in Section 2, we propose ViSparse, a retrieval-based framework that sparsifies visual attention during multimodal reasoning while preserving access to the full visual context throughout the reasoning trajectory. Given N visual tokens $\mathcal{V} = \{v_j\}_{j=1}^{N}$, it stores their complete KV cache in CPU memory during *prefill* and builds head-specific hierarchical cluster indices once prefill is complete. During *decoding*, query-driven retrieval selects a small subset of visual tokens for loading onto the GPU as visual focus evolves across reasoning steps, with each head's refresh frequency adapted to its attention coverage. To exploit similarity across adjacent layers, a leader layer performs retrieval, while its followers reuse the selected positions and load their own layer-specific KV states. We further optimize execution with fused GPU kernels for token selection and gathering, GPU buffer reuse to avoid redundant transfers, and asynchronous cross-layer prefetching to overlap data movement with computation.

3.2 HIERARCHICAL CLUSTERING FOR EFFICIENT RETRIEVAL

To efficiently select the visual tokens relevant to each decoding query, we exploit the similarity among visual-token representations and group them using hierarchical agglomerative clustering. Since these representations vary across attention heads, we perform clustering independently for each head.

Let $\mathbf{v}_j^{(h)}$ denote the representation of visual token v_j at head h. For any cluster $\mathcal{C}$ of visual-token positions, its centroid is the mean of its member representations,

$$\boldsymbol{\mu}^{(h)}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \mathbf{v}_j^{(h)}. \tag{1}$$

Starting from each token as its own cluster, we repeatedly merge two clusters according to the following centroid-based linkage. For two clusters $\mathcal{C}_a$ and $\mathcal{C}_b$, the inter-cluster distance is defined as

$$d(\mathcal{C}_a, \mathcal{C}_b) = \left\| \boldsymbol{\mu}^{(h)}(\mathcal{C}_a) - \boldsymbol{\mu}^{(h)}(\mathcal{C}_b) \right\|_2. \tag{2}$$

At each step, we merge the two clusters whose centroids are closest, with the pair selected as

$$(a^\star, b^\star) \in \arg\min_{a \neq b} d(\mathcal{C}_a, \mathcal{C}_b). \tag{3}$$

A cluster no longer participates in merging once it reaches the predefined token limit. This yields a partition $\{\mathcal{C}_m^{(h)}\}_{m=1}^{M}$ of the visual-token positions in which tokens with similar representations share a cluster, with centroids $\boldsymbol{\mu}_m^{(h)}$. These centroids represent the clusters for query-based scoring, while the cluster assignments map selected clusters to their constituent token positions.

3.3 CLUSTER RETRIEVAL WITH HEAD-ADAPTIVE REFRESH

During decoding, each attention head maintains its own visual-token selection and updates it according to a head-adaptive refresh schedule. When head h refreshes its selection at decoding step t, it scores each cluster centroid against the current query $\mathbf{q}_t^{(h)}$:

$$a_{t,m}^{(h)} = \mathrm{Attn}\left(\mathbf{q}_t^{(h)}, \boldsymbol{\mu}_m^{(h)}\right). \tag{4}$$

We select the highest-scoring clusters until their combined token count reaches the visual-token budget. Let $\mathcal{M}_t^{(h)}$ denote the indices of the selected clusters and $\mathcal{P}_t^{(h)} = \bigcup_{m \in \mathcal{M}_t^{(h)}} \mathcal{C}_m^{(h)}$ the positions of all visual tokens in these clusters. We then load the KV states at these positions onto the GPU for attention.

Beyond ranking clusters, the score mass captured by the active selection indicates how well it covers the head's current visual focus. A high value indicates that the selected clusters capture most relevant content, supporting longer reuse; a low value suggests that relevant content may lie outside the selection, motivating more frequent refreshes. We define this selection coverage as

$$s_{t,h} = \sum_{m \in \mathcal{M}_t^{(h)}} a_{t,m}^{(h)}. \tag{5}$$

Since coverage can fluctuate across queries, we use an exponential moving average (EMA) of coverage to update each head's refresh schedule. The EMA is updated whenever the head refreshes its selection:

$$\widetilde{s}_h \leftarrow \beta\, \widetilde{s}_h + (1-\beta)\, s_{t,h}, \tag{6}$$

where β is the decay factor. We use two refresh intervals, $T_1 < T_2$, and update head assignments every T_2 decoding steps. The fraction $\rho \in (0,1)$ of heads with the lowest EMA coverage refreshes every T_1 steps, while the remaining heads refresh every T_2 steps. Each head reuses its selected tokens between refreshes.

3.4 Sharing Retrieval Indices Across Layers

As shown in Section 2.2, adjacent layers exhibit similar visual-attention patterns, motivating cross-layer sharing of retrieval indices. We determine the sharing groups once using attention collected during prefill. For each layer ℓ, we average the attention weights assigned to each visual token across all text query positions and attention heads. The resulting vector $\mathbf{a}^{(\ell)}$ summarizes how much attention the layer assigns to each visual token. We then compute the cosine similarity between the visual-attention vectors of adjacent layers:

$$\sigma_\ell = \cos\Big(\mathbf{a}^{(\ell)}, \mathbf{a}^{(\ell+1)}\Big), \qquad \ell = 1, \ldots, L-1. \tag{7}$$

A candidate group beginning at layer ℓ and containing g consecutive layers is denoted by

$$B_{\ell,g} = \{\ell, \ell+1, \ldots, \ell+g-1\}, \qquad 2 \le g \le G \le L, \tag{8}$$

where G is the maximum sharing-group size. Our objective is to select K non-overlapping sharing groups that maximize the total similarity between adjacent layers within the selected groups. We formulate this as the following binary linear optimization problem, where $x_{\ell,g} \in \{0,1\}$ indicates whether group $B_{\ell,g}$ is selected:

$$\begin{aligned} \max_{\{x_{\ell,g}\}} \quad & \sum_{g=2}^{G} \sum_{\ell=1}^{L-g+1} \left(\sum_{q=0}^{g-2} \sigma_{\ell+q} \right) x_{\ell,g} \\ \text{s.t.} \quad & \sum_{g=2}^{G} \sum_{\ell=1}^{L-g+1} x_{\ell,g} = K, \qquad \sum_{(\ell,g):\, i \in B_{\ell,g}} x_{\ell,g} \le 1 \quad (i = 1, \ldots, L). \end{aligned} \tag{9}$$

For each selected group, the first layer serves as the leader and the remaining layers serve as followers. Each follower reuses the leader's clustering results, eliminating the need to perform clustering independently. During decoding, followers further reduce retrieval overhead by reusing the leader's selected visual-token positions and refresh schedule while loading the corresponding KV representations from their own layer-specific stores.

3.5 ViSparse System Implementation

We accelerate index construction and visual-token movement with batched tensor operations and fused GPU kernels. During index construction after prefill, we compute distances between cluster centroids in batches across attention heads, reducing Python-loop overhead. We also store the resulting centroids and cluster metadata, including IDs, sizes, and member-token indices, in batched tensors for efficiency. During decoding, fused kernels accelerate cluster selection and the loading of selected visual KV states from CPU to GPU.

Across decoding steps, each head retains $\mathcal{M}_t^{(h)}$, $\mathcal{P}_t^{(h)}$, and the corresponding visual-token data already loaded in the GPU buffer. Let $\mathcal{R}_{t-1}^{(h)}$ denote the resident cluster IDs. At a refresh, the runtime

transfers only the missing set $\Delta\mathcal{M}_t^{(h)} = \mathcal{M}_t^{(h)} \setminus \mathcal{R}_{t-1}^{(h)}$, while reusing resident data otherwise. The GPU buffer also reserves additional space to retain the KV states of recently used but currently unselected clusters for potential reuse. When the buffer is full, LRU eviction removes the least recently used entries, allowing clusters reused in later decoding steps to remain in GPU memory and avoiding redundant CPU-to-GPU transfers.

During prefill, the system determines the cross-layer sharing groups once. For each selected group $B_{\ell,g}$, the inference engine uses the leader's selection to identify each follower's missing clusters before its layer runs and transfers them asynchronously from the follower's CPU store while the preceding layer computes attention. The follower layer then computes attention using the prefetched KV states, allowing data transfer to overlap with computation in the preceding layer.

Table 1: Main accuracy results and average token count. Accuracy is reported in percent; "Avg. acc." is the arithmetic mean over the eight benchmarks, and "Avg. tokens" is the mean token count.

Model	Method	DocVQA	MMCoQA	MMQA	SlideVQA	WebQA	WikiVQA	ChartQA	MMVet	Avg. acc.	Avg. tokens
Qwen3-VL-Thinking-8B	FullKV	97.0	78.0	83.5	92.5	77.0	95.0	77.92	68.85	83.72	1034.03
	SnapKV	96.5	72.0	70.0	71.5	76.5	94.0	60.80	51.42	74.09	2707.98
	SparseMM	96.5	77.0	80.5	90.0	73.5	95.0	74.84	60.73	81.01	1295.56
	LOOK-M	96.5	61.0	70.5	65.5	77.0	95.5	59.00	47.98	71.62	2469.37
	FastV	68.5	49.5	54.0	44.5	72.5	94.0	28.72	29.27	55.12	3247.45
	VisionPulse	96.5	72.5	82.0	91.0	74.0	95.0	75.08	63.81	81.24	1166.62
	HybridKV	**97.0**	74.5	85.0	91.0	75.0	95.5	74.88	63.49	82.05	1109.76
	ViSparse (ours)	96.5	**80.0**	**87.0**	**92.0**	**79.5**	**96.0**	**76.44**	**66.42**	**84.23**	**980.64**
InternVL-3.5-Thinking 4B	FullKV	94.5	70.0	76.0	91.5	66.5	85.0	85.80	70.46	79.97	1730.53
	SnapKV	94.5	70.0	76.5	85.0	66.5	78.5	85.12	59.82	76.99	1737.71
	SparseMM	**95.5**	67.0	75.5	89.0	62.5	83.0	85.48	61.65	77.45	1603.39
	LOOK-M	**95.5**	68.5	74.0	82.0	63.5	84.5	82.76	58.76	76.19	1877.26
	FastV	73.5	55.5	62.0	61.5	58.0	82.5	47.84	49.86	61.34	2199.16
	VisionPulse	95.0	70.5	75.0	87.0	68.0	83.5	85.40	63.49	78.49	1658.66
	HybridKV	94.5	68.0	74.5	86.0	64.0	83.5	**85.68**	58.94	76.89	1536.07
	ViSparse (ours)	**95.5**	**72.5**	**81.0**	**93.5**	**68.5**	**85.5**	85.32	**67.29**	**81.14**	**1514.94**
Qwen3-VL-Thinking-4B	FullKV	96.5	76.0	81.0	91.5	75.5	90.0	78.00	64.13	81.58	1596.49
	SnapKV	95.0	68.0	67.0	68.0	73.0	91.0	58.52	41.15	70.21	3333.09
	SparseMM	96.0	75.5	76.0	87.5	73.5	**92.0**	74.12	51.70	78.29	1775.43
	LOOK-M	95.0	61.0	68.5	63.0	73.0	90.5	55.76	41.65	68.55	3032.29
	FastV	61.5	46.5	48.5	43.5	67.5	88.5	31.56	25.55	51.64	4005.89
	VisionPulse	**96.5**	75.0	76.0	89.5	72.5	90.5	74.80	57.61	79.05	1550.72
	HybridKV	95.0	74.5	78.0	**92.5**	71.0	91.5	71.08	53.76	78.42	1547.13
	ViSparse (ours)	**96.5**	**76.0**	**79.0**	90.0	**75.5**	**92.0**	**75.92**	**63.49**	**81.05**	**1475.64**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and datasets. We evaluate three open-weight multimodal reasoning models: Qwen3-VL-Thinking-8B, Qwen3-VL-Thinking-4B Bai et al. (2025), and InternVL-3.5-Thinking 4B Wang et al. (2025c). Evaluation covers eight benchmarks: DocVQA Mathew et al. (2021), MMCoQA Li et al. (2022), MultiModalQA (MMQA) Talmor et al. (2021), SlideVQA Tanaka et al. (2023), WebQA Chang et al. (2022), WikiVQA Yang et al. (2015), ChartQA Masry et al. (2022), and MMVet Yu et al. (2023). These benchmarks span document and chart understanding, multimodal question answering, and general visual reasoning.

Baselines. We compare ViSparse with FullKV, which retains the complete visual KV cache and attends to all visual tokens, and six visual-token compression methods: SnapKV Li et al. (2024), SparseMM Wang et al. (2025b), LOOK-M Wan et al. (2024), FastV Chen et al. (2024b), VisionPulse Xu et al. (2026), and HybridKV Zeng et al. (2026). These methods reduce the visual context through token pruning, representation merging, or dynamic visual-token selection, providing direct comparisons in accuracy and decoding efficiency.

Implementation details. We use a per-head visual-token retention ratio of $r = 0.1$. The EMA decay is $\beta = 0.9$, and the fraction of heads assigned to the frequent-refresh schedule is $\rho = 0.5$. We set the two refresh intervals to $T_1 = 4$ and $T_2 = 10$ decoding steps. During hierarchical clustering, we cap each cluster at 32 visual tokens. For cross-layer sharing, the maximum group size is $G = 3$ and we select $K = L/6$ disjoint groups. We set the GPU buffer's slack ratio to $S = 0.05$, reserving extra capacity equal to 5% of each head's active visual KV budget. During decoding, we use a

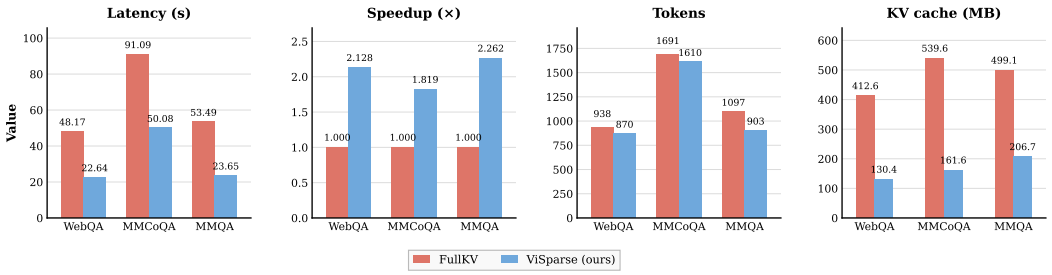


Figure 3: Efficiency comparison between full KV caching and ViSparse on Qwen3-VL-Thinking-8B. Latency is in seconds, speedup is relative to FullKV, tokens are average generated length, and KV cache is peak GPU storage in MB.

temperature of 0.7 and limit the maximum generation length to 16,384 tokens on an NVIDIA H20 GPU.

4.2 Main Results

Table 1 shows that ViSparse achieves the highest average accuracy among compressed methods across all three models. On Qwen3-VL-Thinking-8B and InternVL-3.5-Thinking 4B, it reaches 84.23% and 81.14%, exceeding FullKV by 0.51 and 1.17 percentage points and the strongest compressed baselines by 2.18 and 2.65 points, respectively. Improvements extend across tasks: ViSparse outperforms FullKV on six of eight InternVL benchmarks, with gains on MMCoQA, MMQA, and WebQA also observed for Qwen3-VL-Thinking-8B.

On Qwen3-VL-Thinking-4B, ViSparse achieves 81.05%, only 0.53 percentage points below FullKV and 2.00 points above VisionPulse, the strongest compressed baseline. ViSparse also reduces average generated-token counts relative to FullKV by 5.2%, 12.5%, and 7.6% for Qwen3-VL-Thinking-8B, InternVL-3.5-Thinking 4B, and Qwen3-VL-Thinking-4B, respectively. These results show that visual sparsification preserves competitive accuracy across model families and scales while shortening the generated reasoning traces.

4.3 Efficiency Analysis

We compare the end-to-end cost of full KV caching and ViSparse on the three benchmarks used in the ablations, using Qwen3-VL-Thinking-8B throughout. Figure 3 reports average per-instance latency and generated-token length, speedup, and peak GPU KV-cache memory.

ViSparse achieves latency speedups of 2.128×, 1.819×, and 2.262× on WebQA, MMCoQA, and MultiModalQA, respectively. The average generated-token lengths remain comparable to FullKV, changing from 937.57 to 869.77 on WebQA, from 1690.84 to 1610.19 on MMCoQA, and from 1096.57 to 902.77 on MultiModalQA. The consistent latency gains therefore primarily reflect more efficient visual-attention computation and data movement rather than shorter decoding traces. Peak GPU KV-cache memory falls from 412.6 to 130.4 MB on WebQA, from 539.6 to 161.6 MB on MMCoQA, and from 499.1 to 206.7 MB on MultiModalQA, corresponding to reductions of 68.4%, 70.1%, and 58.6%, respectively. These results show that the method reduces both the per-step visual-attention cost and the memory required to hold the active visual context.

4.4 Ablation Experiments

Our ablation study examines the visual-token retention ratio, clustering rule, head assignment, assignment updates, and cross-layer index sharing. Unless stated otherwise, we use the Section 4.1 configuration with Qwen3-VL-Thinking-8B; accuracy is reported in percent and latency in seconds.

Visual-token retention ratio r. Figure 4 varies the fraction of visual tokens retained by each head. The default $r = 10\%$ gives the highest average accuracy (82.17%), whereas 5% reaches 78.00%.

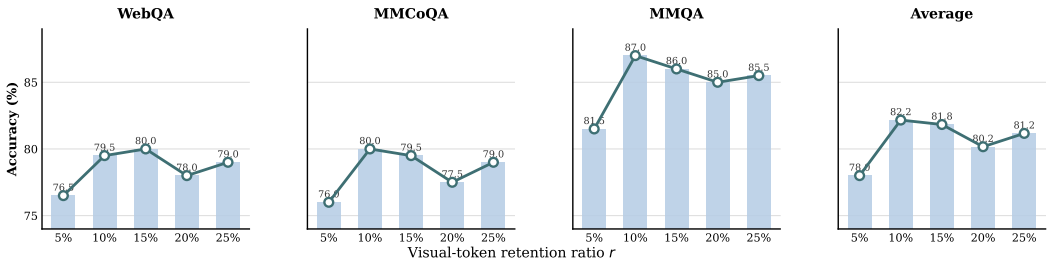


Figure 4: Effect of the visual-token retention ratio r under compressed settings.

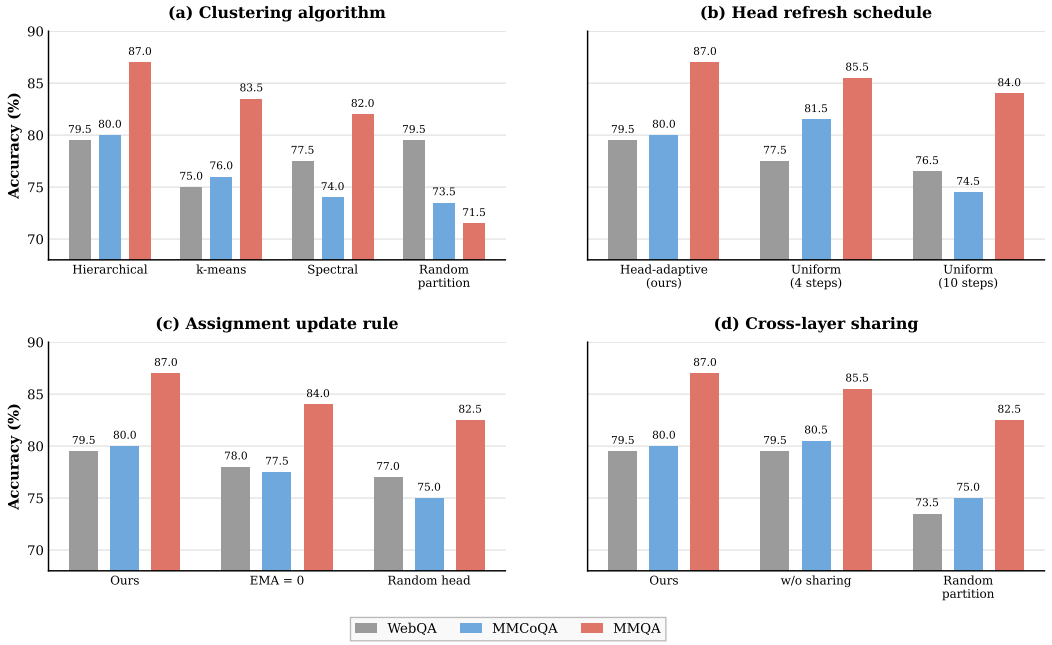


Figure 5: Component ablations on Qwen3-VL-Thinking-8B: (a) clustering, (b) refresh schedule, (c) assignment update, and (d) cross-layer index sharing.

Increasing the budget beyond 10% does not improve the average: the 15%, 20%, and 25% settings obtain 81.83%, 80.17%, and 81.17%, respectively. All settings at or above 10% maintain high accuracy, with averages between 80.17% and 82.17%, while increasing the budget does not yield a consistent improvement. These results suggest that a moderate retention budget is sufficient to preserve the visual evidence needed for accurate reasoning.

Clustering algorithm. We replace ViSparse's hierarchical clustering with alternative cluster assignments (Figure 5(a)). Hierarchical clustering gives the highest average accuracy (82.17), outperforming k-means, spectral clustering, and random partitioning by 4.00, 4.33, and 7.33 points, respectively. Random partitioning is especially weak on MultiModalQA and MMCoQA (71.5% and 73.5%), with the lowest overall average (74.83%). These results support the use of hierarchical clustering in ViSparse, which achieves the highest average accuracy among the evaluated partitioning methods under the same cluster count and retention ratio.

Head assignment. Figure 5(b) compares ViSparse's head-adaptive refresh schedule with uniform refresh every 4 or 10 decoding steps. ViSparse achieves the highest average accuracy (82.17%), versus 81.50% and 78.33%, respectively. Uniform 4-step refresh improves MMCoQA accuracy but

lowers it on WebQA and MultiModalQA, while uniform 10-step refresh degrades all three benchmarks. These results support adapting refresh intervals to each head's selection coverage, maintaining accuracy with fewer refreshes than the uniform 4-step schedule.

Assignment update. Figure 5(c) examines the roles of temporal smoothing and coverage-aware head assignment. Setting the EMA decay to zero lowers average accuracy from 82.17 to 79.83, since the assignment then depends only on the latest coverage observation, which may reflect a transient change in the decoding query. Randomly assigning heads to the refresh schedules further reduces the average accuracy to 78.17, with drops of 2.5, 5.0, and 4.5 points on WebQA, MMCoQA, and MultiModalQA, respectively. These results suggest that head-adaptive refresh benefits from both coverage-aware interval assignment and temporal smoothing of the coverage signal.

Cross-layer sharing. Figure 5(d) compares the proposed layer grouping with no sharing and random grouping. Grouping layers by attention similarity preserves accuracy while allowing retrieval indices to be reused: average accuracy is 82.17%, compared with 81.83% without sharing. Random grouping reduces accuracy on all three benchmarks, yielding an average of 77.00%. These results suggest that attention similarity provides a useful criterion for identifying layers that can share retrieval indices without compromising accuracy.

5 Related Work

Multimodal Reasoning. Building on the success of chain-of-thought reasoning in improving performance on complex tasks with large language models by scaling inference-time computation Wei et al. (2022); Jaech et al. (2024), this paradigm was rapidly transferred to multimodal LLMs (MLLMs), shifting their target from perception-and-description to tasks that require multi-step inference over visual evidence Wang et al. (2024; 2025d), including mathematical and geometric problem solving Huang et al. (2026); Lan et al. (2025), chart and document understanding Xu et al. (2025); Jia et al. (2025), medical diagnosis and analysis Das et al. (2026); Koleilat et al. (2026), and spatial, 3D, and temporal reasoning Zhao et al. (2026); Zhang et al. (2026b), with such capabilities instilled through SFT Zhang et al. (2025); Qiu et al. (2025) or RL Chen et al. (2025); Jain et al. (2026). These gains, however, come with substantial memory and computational costs: visual tokens dominate the context of high-resolution and video inputs Zhang et al. (2026a), placing considerable pressure on GPU memory, while long reasoning chains compound the computational cost by repeatedly attending to these tokens over many decoding steps.

Token Compression. High-resolution images and multi-frame videos introduce thousands of visual tokens, increasing attention computation and KV-cache memory requirements Yu et al. (2026); Ye et al. (2026); Shao et al. (2025). This motivates visual token compression for efficient multimodal inference. Some methods reduce visual redundancy by exploiting similarities and diversity among token representations Wang et al. (2025a); Yang et al. (2025); Alvar et al. (2025); Jiang et al. (2025), while others use attention scores to identify and retain important tokens Liu et al. (2024); Zhang et al. (2024); He et al. (2024); Shao et al. (2026). VisionPulse Xu et al. (2026) selects visual tokens at each decoding step using attention from a single anchor layer. HybridKV Zeng et al. (2026) classifies heads during prefill and combines static pruning with chunk-wise dynamic retrieval. ViSparse preserves the complete visual KV cache in CPU memory and combines head-specific hierarchical cluster retrieval, coverage-adaptive refresh, and cross-layer index reuse.

6 Conclusion

We presented ViSparse, a retrieval-based framework for efficient visual attention in multimodal reasoning. It retains the complete visual KV cache in CPU memory while using head-specific hierarchical cluster retrieval, coverage-adaptive refresh, and cross-layer reuse to load only the visual tokens needed at each decoding step. Across three multimodal reasoning models and eight benchmarks, ViSparse maintains accuracy close to FullKV and consistently outperforms the evaluated compressed baselines, while reducing generated-token counts and the cost of visual-context processing.

AI USE STATEMENT

During the preparation of this manuscript, we used generative AI tools as editorial assistance to improve grammar, phrasing, clarity, and readability, and to help identify inconsistencies in the presentation. The tools were not used to generate experimental data or results. All AI-assisted suggestions and edits were reviewed and verified by the authors.

ETHICS STATEMENT

This work does not involve human participants, human-subject experiments, or the collection of new personal data. We evaluate ViSparse using publicly available benchmark datasets and open-weight multimodal language models, in accordance with their respective licenses and terms of use. Our work focuses on improving inference efficiency and does not introduce a new high-stakes application.

REPRODUCIBILITY STATEMENT

Detailed descriptions of ViSparse and its implementation are provided in the methodology and experimental sections. The paper specifies the hierarchical retrieval algorithm, head-adaptive refresh policy, cross-layer index sharing, GPU buffering, and asynchronous prefetching, with pseudocode provided in the appendix. The experimental setup lists the evaluated models, public benchmarks, baselines, hyperparameters, decoding settings, and hardware platform. Together, these materials provide the information needed to reproduce the reported accuracy, latency, and memory measurements.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

I Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, Chun-Yi Lee, et al. Retraining-free merging of sparse moe via hierarchical clustering. *arXiv preprint arXiv:2410.08589*, 2024a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024b.

Zhangquan Chen, Xufang Luo, and Dongsheng Li. Visrl: Intention-driven visual perception via reinforced reasoning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2545–2555. IEEE, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Aryan Das, Tanishq Rachamalla, Koushik Biswas, Swalpa Kumar Roy, and Vinay Kumar Verma. Uncertainty-aware vision-language segmentation for medical imaging. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 8490–8499. IEEE, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification. *arXiv preprint arXiv:2410.08584*, 2024.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jitesh Jain, Jialuo Li, Zixian Ma, Jieyu Zhang, Chris Dongjoo Kim, Sangho Lee, Rohun Tripathi, Tanmay Gupta, Christopher Clark, Humphrey Shi, et al. Sage: Training smart any-horizon agents for long video reasoning with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41478–41488, 2026.

Caijun Jia, Nan Xu, Jingxuan Wei, Qingli Wang, Lei Wang, Bihui Yu, and Junnan Zhu. Chartreasoner: Code-driven modality bridging for long-chain reasoning in chart question answering. *arXiv preprint arXiv:2506.10116*, 2025.

Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4075–4083, 2025.

Taha Koleilat, Hojat Asgariandehkordi, Omid Nejatimanzari, Berardino Barile, Yiming Xiao, and Hassan Rivaz. Medclipseg: probabilistic vision-language adaptation for data-efficient and generalizable medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1406–1417, 2026.

Xiaohan Lan, Fanfan Liu, Haibo Qiu, Siqi Yang, Delian Ruan, Peng Shi, and Lin Ma. Metis-home: Hybrid optimized mixture-of-experts for multimodal reasoning. *arXiv preprint arXiv:2510.20519*, 2025.

Yongqi Li, Wenjie Li, and Liqiang Nie. 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)*, pp. 4220–4231, 2022.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Haibo Qiu, Xiaohan Lan, Fanfan Liu, Xiaohu Sun, Delian Ruan, Peng Shi, and Lin Ma. Metis-rise: Rl incentivizes and sft enhances multimodal reasoning model learning. *arXiv preprint arXiv:2506.13056*, 2025.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models. *arXiv preprint arXiv:2507.20198*, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 38:135547–135570, 2026.

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

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

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

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4065–4078, 2024.

Han Wang, Yuxiang Nie, Yongjie Ye, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-vlm: Simple dynamic visual token compression for videollm. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20812–20823. IEEE, 2025a.

Jiahui Wang, Zuyan Liu, Yongming Rao, and Jiwen Lu. Sparsemm: Head sparsity emerges from visual concept responses in mllms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23177–23187. IEEE, 2025b.

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

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025d.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Hengbo Xu, Shengjie Jin, Yanbiao Ma, and Zhiwu Lu. Visionpulse: Dynamic visual sparsity for efficient multimodal reasoning. *arXiv preprint arXiv:2605.31457*, 2026.

Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, Chengjin Xu, Chun Yuan, and Jian Guo. Chartmoe: Mixture of diversely aligned expert connector for chart understanding. In *International Conference on Learning Representations*, volume 2025, pp. 78550–78572, 2025.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 2013–2018, 2015.

Haotian Ye, Qiyuan He, Jiaqi Han, Puheng Li, Jiaojiao Fan, Zekun Hao, Fitsum Reda, Yogesh Balaji, Huayu Chen, Sheng Liu, et al. Infotok: Adaptive discrete video tokenizer via information-theoretic compression. In *International Conference on Learning Representations*, volume 2026, pp. 40047–40069, 2026.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free mllm acceleration. *arXiv preprint arXiv:2601.22674*, 2026.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Bowen Zeng, Feiyang Ren, Jun Zhang, Xiaoling Gu, Ke Chen, Lidan Shou, and Huan Li. Hybridkv: Hybrid kv cache compression for efficient multimodal large language model inference. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13018–13034, 2026.

Jun Zhang, Yicheng Ji, Feiyang Ren, Yihang Li, Bowen Zeng, Zonghao Chen, Ke Chen, Lidan Shou, Gang Chen, and Huan Li. Efficient inference for large vision-language models: Bottlenecks, techniques, and prospects. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21036–21066, 2026a.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1662, 2025.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

Zaibin Zhang, Yuhan Wu, Lianjie Jia, Yifan Wang, Zhongbo Zhang, Yijiang Li, Binghao Ran, Fuxi Zhang, Zhuohan Sun, Zhenfei Yin, et al. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*, 2026b.

Haoyu Zhao, Akide Liu, Zeyu Zhang, Weijie Wang, Feng Chen, Ruihan Zhu, Gholamreza Haffari, and Bohan Zhuang. Cov: Chain-of-view prompting for spatial reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32427–32440, 2026.

A Component Runtime Breakdown

Table 2 reports the runtime shares of ViSparse's components on Qwen3-VL-Thinking-8B, averaged across datasets.

Table 2: Component runtime shares for Qwen3-VL-Thinking-8B, averaged across datasets.

Component	Runtime share (%)
Head-specific hierarchical clustering	0.61
Coverage estimation and head assignment	0.30
Visual KV offloading and GPU buffer allocation	0.23
Cluster retrieval	1.55
Visual KV loading	2.30
Total	4.99

The five components together account for 4.99% of runtime. Before decoding, ViSparse offloads the complete visual KV cache from GPU to CPU memory and allocates GPU buffers, together accounting for 0.23% of runtime. During decoding, cluster retrieval selects relevant visual tokens (1.55%), and visual KV loading transfers their KV states from CPU memory to the GPU only when they are

not already in the GPU buffer (2.30%). The remaining 0.91% comes from head-specific hierarchical clustering (0.61%) and coverage estimation and head assignment (0.30%). Cluster retrieval and visual KV loading together account for 3.85% of runtime, making them the main targets for further optimization.

B ViSparse Performance in Video Reasoning

To assess ViSparse in video reasoning, we report results on Video-MME Fu et al. (2025), a benchmark covering short, medium, and long videos. As shown in Table 3, ViSparse stays within one percentage point of FullKV across all video-duration groups, with an overall gap of only 0.7 percentage points. The close agreement across short, medium, and long videos indicates that retrieval-based visual sparsification preserves video-reasoning accuracy even as the temporal context becomes longer.

Table 3: ViSparse performance on video inputs, evaluated on the Video-MME benchmark. Accuracy and average generated-token count are reported for each duration category and overall.

Method	Short		Medium		Long		Overall	
	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens	Accuracy	Tokens
FullKV	0.782	447.97	0.668	618.44	0.573	742.86	0.674	603.09
ViSparse	0.774	410.42	0.662	590.38	0.563	780.55	0.667	593.78

C Additional Visualizations

Figure 6 supplements the examples in Section 2.1 with additional heads and decoding steps. The top row shows Heads 4–7 at step 1, while the middle and bottom rows show Heads 1–4 at steps 25 and 75, respectively. Across heads and decoding steps, the maps exhibit both localized hotspots and more broadly distributed responses, with the attended regions changing over the reasoning trajectory. These examples further support the observation in the main text that visual attention varies across heads and reasoning steps.

D Limitations and Future Directions

Our efficiency evaluation focuses on the NVIDIA H20 platform. Because ViSparse retrieves visual KV states from CPU memory, the measured latency reflects the bandwidth and compute characteristics of this setup. Extending the evaluation to additional hardware platforms and serving configurations would further characterize the portability and practical performance of the implementation.

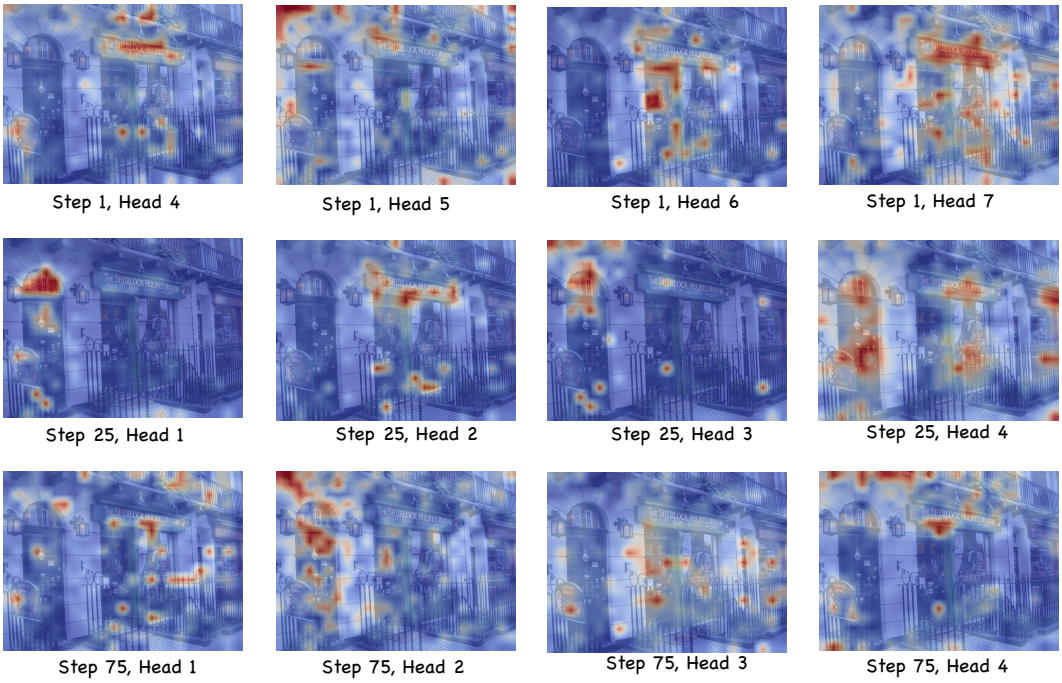


Figure 6: Additional visual-attention maps. The top row shows Heads 4–7 at decoding step 1; the middle and bottom rows show additional attention maps at steps 25 and 75, respectively. Warmer colors indicate higher attention weights.

E ALGORITHM PSEUDOCODE

Algorithm 1 summarizes ViSparse using the notation in the main text, with layer indices added explicitly. Here, $\tau_{\ell,h}$ is the last refresh step and $\mathcal{U}_{\ell,h}$ contains the visual-token positions covered by clusters resident in the GPU buffer. For a follower head h in layer ℓ, the corresponding leader head is the head with the same index h in the leader layer. The follower reuses the leader's cluster indices, selected positions, and refresh schedule while retaining its own KV states. Standard transformer operations are omitted.

Algorithm 1 ViSparse inference

Require: Multimodal input; retention ratio r; EMA decay β; frequent-refresh fraction ρ; intervals $T_1 < T_2$
Ensure: Generated output sequence

Run prefill and store the complete visual KV cache in CPU memory
Determine sharing groups and their leader layers; allocate GPU buffers
Build head-specific clusters $\mathcal{C}_{\ell,m}^{(h)}$ and centroids $\boldsymbol{\mu}_{\ell,m}^{(h)}$ for each leader and ungrouped layer
Let each follower reuse the cluster indices and centroids of its leader layer at the same head indices
Initialize $\mathcal{P}_{\ell}^{(h)} \leftarrow \emptyset, \mathcal{U}_{\ell,h} \leftarrow \emptyset, \widetilde{s}_{\ell,h} \leftarrow 0, T_{\ell,h} \leftarrow T_1$, and $\tau_{\ell,h} \leftarrow 0$
for each decoding step t **do**
  for each layer ℓ in execution order **do**
    Compute the current queries $\mathbf{q}_{\ell,t}^{(h)}$ for all heads
    for each head h **do**
      if ℓ is a follower with leader layer ℓ' **then**
        $\mathcal{P}_{\ell}^{(h)} \leftarrow \mathcal{P}_{\ell'}^{(h)}, T_{\ell,h} \leftarrow T_{\ell',h}$, and $\tau_{\ell,h} \leftarrow \tau_{\ell',h}$
      else if $\mathcal{P}_{\ell}^{(h)} = \emptyset$ or $t - \tau_{\ell,h} \geq T_{\ell,h}$ **then**
        $a_m \leftarrow \mathrm{Attn}(\mathbf{q}_{\ell,t}^{(h)}, \boldsymbol{\mu}_{\ell,m}^{(h)}), m = 1, \ldots, M$
        Select clusters $\mathcal{M}_{\ell}^{(h)}$ by descending a_m until the token budget rN is reached
        $\mathcal{P}_{\ell}^{(h)} \leftarrow \bigcup_{m \in \mathcal{M}_{\ell}^{(h)}} \mathcal{C}_{\ell,m}^{(h)}; s \leftarrow \sum_{m \in \mathcal{M}_{\ell}^{(h)}} a_m$
        if $\tau_{\ell,h} = 0$ **then**
          $\widetilde{s}_{\ell,h} \leftarrow s$
        else
          $\widetilde{s}_{\ell,h} \leftarrow \beta \widetilde{s}_{\ell,h} + (1 - \beta) s$
        end if
        $\tau_{\ell,h} \leftarrow t$
      end if
      $\mathcal{D}_{\ell,h} \leftarrow \mathcal{P}_{\ell}^{(h)} \setminus \mathcal{U}_{\ell,h}$ ▷ Missing token positions
      Load KV states at $\mathcal{D}_{\ell,h}$ from this layer's CPU store; reuse resident states
      Evict least-recently-used resident clusters only when needed; update $\mathcal{U}_{\ell,h}$
    end for
    if ℓ is a group leader **then**
      Launch asynchronous prefetch of followers' selected KV states from their own CPU stores
    end if
    Wait for the current layer's required transfers and assemble its selected visual KV states
    Compute attention with the selected visual tokens and all text and generated tokens
  end for
  if $t = 1$ or $t \bmod T_2 = 0$ **then**
    In each leader or ungrouped retrieving layer, assign T_1 to the fraction ρ of heads with lowest $\widetilde{s}_{\ell,h}$, and T_2 to the remaining heads
    Followers inherit the corresponding leader head's interval and refresh schedule
  end if
  Generate the next token; stop at EOS or the generation limit
end for