

SparseVILA: Decoupling Visual Sparsity for Efficient VLM Inference

Samir Khaki⁴ Junxian Guo² Jiaming Tang² Shang Yang² Yukang Chen¹
 Konstantinos N. Plataniotis⁴ Yao Lu¹ Song Han^{1,2} Zhijian Liu^{1,3}

¹NVIDIA ²MIT ³UC San Diego ⁴University of Toronto

Abstract: Vision Language Models (VLMs) have rapidly advanced in integrating visual and textual reasoning, powering applications across high-resolution image understanding, long-video analysis, and multi-turn conversation. However, their scalability remains limited by the growing number of visual tokens that dominate inference latency. We present **SparseVILA**, a new paradigm for efficient VLM inference that *decouples* visual sparsity across the prefilling and decoding stages. SparseVILA distributes sparsity across stages by pruning redundant visual tokens during prefill and retrieving only query-relevant tokens during decoding. This decoupled design matches leading prefill pruning methods while preserving multi-turn fidelity by retaining most of the visual cache so that query-aware tokens can be retrieved at each conversation round. Built on an AWQ-optimized inference pipeline, SparseVILA achieves up to **4.0× faster prefilling**, **2.5× faster decoding**, and an overall **2.6× end-to-end speedup** on long-context video tasks – while improving accuracy on document-understanding and reasoning tasks. By decoupling query-agnostic pruning and query-aware retrieval, SparseVILA establishes a new direction for efficient multimodal inference, offering a training-free, architecture-agnostic framework for accelerating large VLMs without sacrificing capability.

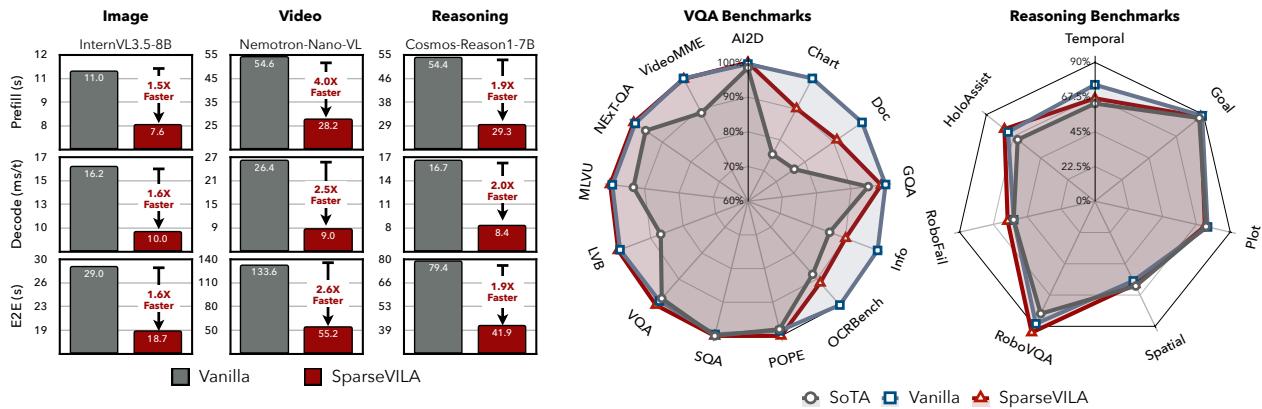


Figure 1 | **Figure 1 | SparseVILA – Efficient VLM Inference.** (a) SparseVILA delivers consistent speedups across image, video, and reasoning tasks, with up to **4.0×** gains in the prefill stage, **2.5×** gains in decoding throughput, and **2.6×** end-to-end speedup. (b) SparseVILA maintains competitive accuracy across most VQA and reasoning benchmarks. While document understanding tasks show a modest drop, performance remains above **90%** of the best reported scores, whereas other compression methods often fall below **75%**. Inference speed in (a) is measured using a single NVIDIA A6000 GPU. Accuracy values in (b) are normalized relative to the highest score for each benchmark.

1. Introduction

Vision Language Models (VLMs) have emerged as a state-of-the-art conversational tool enabling users to directly interact with Large Language Models (LLMs) using various visual features, including photographs, documents, and videos [1, 2, 3, 4, 5]. Unfortunately, this added modality comes at the expense of higher latency and memory associated with processing the visual tokens in the LLM. Hence, deploying LLMs efficiently at

inference time remains a challenge.

Several works have aimed to reduce these associated costs through model pruning on the LLM or vision encoder [6, 7, 8], KV cache compression [9, 10, 11], and most recently, token sparsification [12, 13]. By reducing the amount of computation in the inference pipeline, many of these methods can achieve significant context stage savings, as this stage of the network is mainly compute-bound.

Looking beyond the context, real-world applications often demand extensive generation. Tasks such as image captioning may require a few hundred tokens; meanwhile, video captioning or detailing easily requires more than a few thousand generated tokens. Hence, in such applications, it is not sufficient to only focus on context stage optimization – efficient implementations with real-world applications should focus on both context and decoding stage optimizations. One such example is a multi-turn conversation.

Multi-turn/round conversation serves as a practical use case for VLMs, wherein a user may pose multiple questions about a given visual input. In fact, most benchmarks inherently support multi-round conversation: the GQA dataset [14] has more than 90 questions for the same visual input. Despite this, most evaluation benchmarks run single-round evaluation (*i.e.*, repetitive pre-filling), which is not only unrealistic but also inefficient, as the context stage would be repeated for each generation round. In real-world scenarios, the visual input could span tens of thousands of context tokens; hence, repetitive pre-filling would dramatically slow down the user’s interaction with their VLM.

In this work, we aim to present a unified approach for tackling context and decoding latency in modern VLMs. Latency is a common challenge associated with VLMs due to the sheer amount of visual tokens to be processed. Existing methods for accelerating VLM inference primarily focus on token-wise pruning or merging techniques, with recent approaches leveraging textual priors to reduce visual token complexity in a query-aware manner. In practice, methods that permanently remove visual tokens during the context stage are quite lossy in multi-turn evaluations, as visualized in Figure 1. Hence, in this paper, we introduce SparseVILA as a novel approach for accelerating VLM inference, while retaining multi-turn performance.

Our key insight is a *decoupled sparsity framework* enabling SparseVILA to migrate sparsity from the prefill into the decoding stage. Further, SparseVILA leverages query-aware retrieval in the decoding stage, supporting multi-turn conversation as a different subset of context tokens can be retrieved per question. This *decoupled* approach allows SparseVILA to achieve significant performance improvements in image-centric benchmarks, as shown in Figure 1 and outperforms previous methods in long-context/generation scaling.

2. Preliminaries

Token pruning has proven effective in accelerating inference across a variety of tasks, including image classification [15, 16, 17], object detection [18], and semantic segmentation [18, 19]. With the rise of generative AI, these techniques have been further extended to diffusion models [20], large language models [21], and vision-language models [22, 13, 23]. We refer the readers to Section 5 for a detailed survey of related work.

This paper focuses on token pruning/sparsity for vision-language models (VLMs). The key idea is that *not all visual tokens contribute equally* to VLM’s final prediction. By identifying and removing less informative tokens, it is possible to significantly reduce the computational cost of VLM inference and thereby improve efficiency. Existing work in this area largely differs in how these tokens are selected. We categorize prior methods into two groups based on their dependence on the input query: (i) *query-agnostic* approaches, which identify unimportant tokens based solely on visual saliency or redundancy, and (ii) *query-aware* approaches, which incorporate the semantic relationship between visual and textual inputs to guide pruning. For each category, we highlight representative methods and their limitations, laying the groundwork for our method.

2.1. Query-Agnostic Sparsity

Query-agnostic token pruning methods aim to reduce redundancy or select important visual tokens without relying on the textual input (*i.e.*, query). They prune tokens based solely on the visual context, either within or after the vision encoder. For example, PruMerge [24] clusters and discards less informative tokens using final-layer attention scores, while VisionZip [13] employs a token merging module, similar to ToMe [15], to compress redundant visual information.

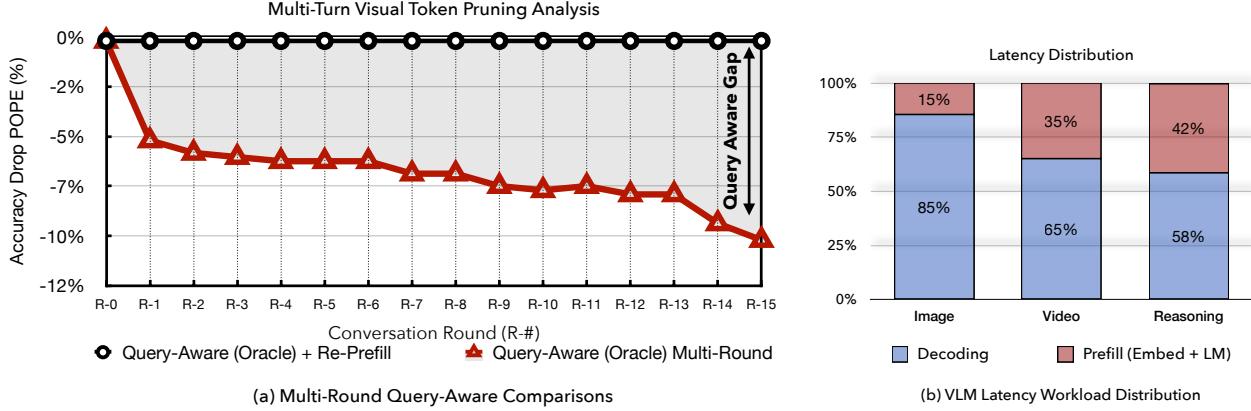


Figure 2 | Multi-Round Query-Aware Comparisons with LLaVA-1.5 [1] on the POPE [25] dataset. Without re-prefilling the context, the query-aware oracle degrades heavily, indicating the inability of query-aware pruning to scale effectively in a multi-turn conversation.

Figure 3 | Latency distribution over the prefilling and decoding stages across image, video, and reasoning workloads.

This visual-only focus, however, presents key limitations. First, these methods often sacrifice fine-grained visual details, especially under high sparsity, which can degrade performance. More importantly, they cannot adapt token selection based on the input query, leading to suboptimal results when task-relevant information is sparsely distributed. By treating all visual tokens uniformly, they risk discarding critical information necessary for accurate reasoning.

2.2. Query-Aware Sparsity

Query-aware pruning improves visual token selection by explicitly modeling the relationship between textual queries and visual representations. For example, FastV [12] leverages attention maps from early LLM layers as salience indicators to guide token pruning during the prefill stage, while SparseVLM [22] uses query-to-vision attention to discard less relevant visual tokens.

Although effective for single-turn tasks, query-aware pruning faces notable limitations in multi-turn interactions. Pruning decisions made for an initial query can permanently remove visual information crucial for subsequent questions, leading to degraded performance across conversation rounds. Empirically, such methods show sharp accuracy drops in multi-turn dialogue, often underperforming even query-agnostic baselines.

To examine this limitation, we construct a *query-aware oracle* that greedily selects an optimal subset of visual tokens to maximize agreement with the unpruned model’s responses. The oracle represents the theoretical upper bound for any query-aware approach, as it directly leverages both the current query and the ground-truth response during selection. Yet, as shown in Figure 2, even this oracle exhibits substantial degradation over successive conversation rounds, highlighting a fundamental constraint of query-dependent pruning: once informative tokens are removed, they cannot be recovered in later turns. These findings motivate the need for a decoupled sparsity framework that preserves visual coverage during prefill while allowing query-aware retrieval during decoding.

3. SparseVILA: Best of Both Worlds

Query-agnostic and query-aware pruning offer complementary benefits but also have inherent limitations. Query-agnostic methods efficiently remove redundant visual tokens without requiring text input, making them stable across multi-turn dialogue, yet they fail to adapt to query-specific relevance. In contrast, query-aware methods dynamically align visual attention with the current query, improving single-turn reasoning, but suffer from irreversible information loss and degraded performance in later conversation rounds once tokens are pruned. These opposing trade-offs motivate our approach. The key insight is that visual sparsity should not be applied uniformly across the inference pipeline. Instead, it should adapt to the distinct roles of the *prefill* and *decoding* stages: the former constructs the multimodal context once, while the latter dominates overall



Figure 4 | Overview of SparseVILA’s decoupled sparsity framework. In the prefill stage, query-agnostic pruning removes redundant visual tokens based on salience scores from the visual encoder, yielding a compact representation shared across conversation turns. During decoding, query-aware retrieval selects only the most relevant visual tokens from the KV cache for attention computation, accelerating generation while maintaining multi-turn fidelity.

latency during iterative generation.

In this paper, we introduce SparseVILA, a framework that achieves the *best of both worlds* by **decoupling** visual compression across the two stages. SparseVILA performs lightweight, query-agnostic pruning during prefill to reduce redundancy without sacrificing coverage, and applies aggressive, query-aware retrieval during decoding when the question is known. This design also better matches the decoding-heavy latency profile of modern VLMs (see Figure 3), yielding significant speedups while maintaining high accuracy across image, video, and reasoning tasks. By separating *when* and *how* sparsity is applied, SparseVILA preserves contextual grounding for future turns and enables efficient, query-conditioned reasoning.

3.1. Prefill Phase: Query-Agnostic Pruning

During the prefill stage, the vision–language model (VLM) encodes the system prompt, visual tokens, and optionally the first user query to construct the multimodal context. To ensure stable performance across multiple dialogue turns, pruning at this stage must remain strictly *query-agnostic*—guided only by visual redundancy or salience rather than any text-conditioned correlation. Since the visual context is computed once and reused throughout the conversation, pruning must retain sufficient coverage for future queries while minimizing redundant information.

Token Salience Estimation. We estimate token importance directly from the visual encoder’s self-attention maps, providing a query-independent measure of visual salience. Following prior work [13, 24, 26], we aggregate attention signals to quantify each token’s contribution to the overall representation, pruning those with the lowest aggregate salience. For models with a single summary token (e.g., CLIP), salience is defined by each token’s attention contribution to this global embedding. For encoders such as RADIO [27, 28], which employ multiple summary tokens, we compute salience as the mean attention directed toward these summary tokens, effectively capturing the same global aggregation behavior. For models without summary tokens (e.g., SigLIP, QwenVL), importance is estimated by averaging intra-visual attention across all tokens.

Efficient Implementation. For long-context inputs such as video sequences, attention-based salience estimation can be memory- and latency-intensive. To address this, we implement a custom Triton [29] kernel that streams softmax normalization and salience accumulation without explicitly forming the full attention matrix. This enables efficient salience computation even for hundreds of thousands of tokens. Empirically, the kernel yields up to a **3 \times** acceleration for SigLIP-style encoders and up to **10 \times** for QwenVL-style encoders (Figure 5a), forming the computational foundation for SparseVILA’s scalable prefill pruning.

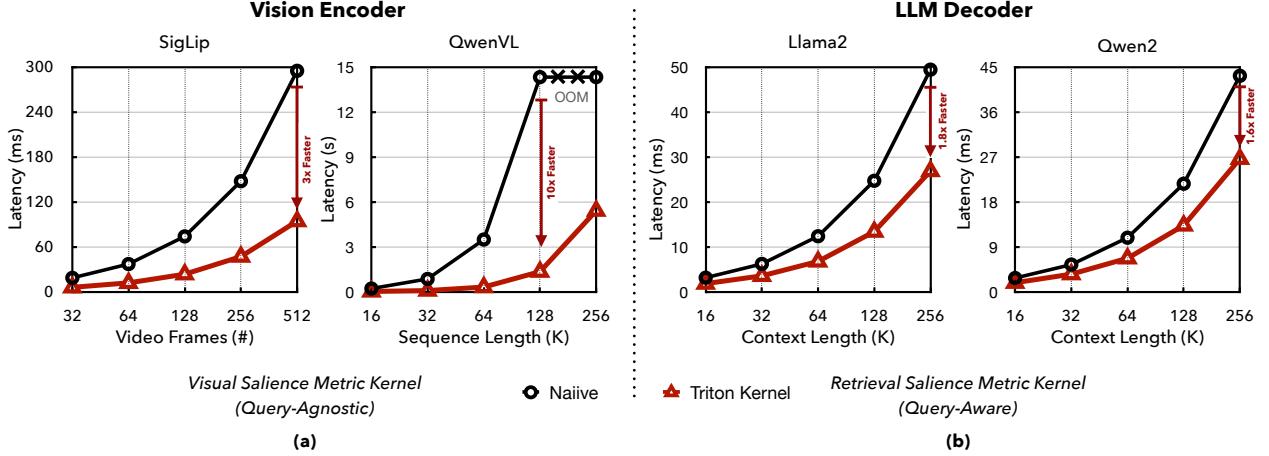


Figure 5 | Salience Metric Kernels. Latency comparison between the naïve and custom Triton implementations across two settings: (a) query-agnostic salience computation for the SigLIP and QwenVL vision encoders, and (b) query-aware retrieval salience for the Llama2 and Qwen2 decoder backbones. Our custom kernels consistently accelerate both query-agnostic and retrieval salience computations, achieving up to 10× and 1.8× speedups, respectively.

3.2. Decode Phase: Query-Aware Retrieval

During the decoding phase, the VLM becomes memory-bound as it repeatedly computes next-token predictions using the pre-filled KV cache. To accelerate this process, SparseVILA selectively activates only the most query-relevant visual tokens during decoding attention, while preserving the rest of the visual information in the KV cache for potential use in later turns. This design enables query-conditioned sparsity without permanently discarding context, maintaining the flexibility required for multi-turn reasoning.

Query-Aware Token Selection. Before decoding begins, SparseVILA estimates the relevance of each visual token to the current query using attention-based salience. Specifically, it measures the aggregate attention strength between the query embeddings and visual entries in the KV cache, providing a query-aware signal that highlights which tokens the model is most likely to reference during generation. Tokens with the highest relevance scores are retained for decoding, while less relevant tokens remain cached but inactive. This dynamic retrieval process effectively narrows the attention scope to the most informative subset of visual tokens, improving efficiency without compromising context consistency. We extend the Triton kernel from the prefill stage to stream the relevance computation directly between the query and cached visual tokens. This operation executes concurrently with the FlashAttention2 [30] path during prefill, yielding up to a **1.5×** speedup over a naïve implementation (Figure 5b). Once salience scores are obtained, the selected visual KV entries are compactly packed into a contiguous memory region, avoiding irregular sparse access patterns during autoregressive decoding.

Rotary Embeddings. Modern VLMs employ rotary position embeddings (RoPE) to encode positional information across modalities. Conventional architectures such as LLaVA-NeXT and LongVILA apply a unified RoPE to both text and vision tokens, while newer models like Qwen2.5-VL use multimodal RoPE to maintain distinct positional grids. When pruning tokens across prefill and decoding stages, these positional structures can become misaligned. For models using unified RoPE, we simply retain a contiguous range of position indices corresponding to the preserved visual tokens. For multimodal RoPE, we reconstruct the minimal contiguous positional grid along temporal, height, and width dimensions and then shift subsequent text positions to maintain global continuity. This adjustment ensures consistent cross-modal alignment even under aggressive token compression, preserving the integrity of the shared positional embedding space across the entire KV cache.

3.3. Decoupled Prefill–Decode Visual Sparsity

SparseVILA introduces a decoupled sparsity framework that explicitly separates *where* and *how* visual compression is applied across the inference pipeline. This design is motivated by the distinct computational characteristics of the two stages: the *prefill stage* executes once per visual input to build the multimodal context, while the *decoding stage* performs iterative next-token prediction and typically dominates end-to-end latency (Fig. 2). Applying uniform sparsity across both is therefore suboptimal—aggressive prefill pruning can permanently discard visual information required for later turns, whereas decoding remains the primary runtime bottleneck.

To address this imbalance, SparseVILA decouples sparsity between stages: lightweight, query-agnostic pruning is applied during prefill to remove globally redundant tokens while retaining sufficient visual coverage, and aggressive, query-aware retrieval is applied during decoding to focus computation on the most relevant visual cues. This adaptive allocation introduces sparsity where it yields the greatest efficiency gain, without compromising contextual grounding for future queries.

We compare the decoupled design with a *prefill-only* sparsity baseline on RoboVQA [31] (Table 1). When tuned for equivalent end-to-end speedup, reallocating sparsity toward decoding consistently improves task performance. The prefill stage retains enough visual tokens to maintain context integrity, while decoding sparsity effectively targets the dominant latency source in multimodal generation.

Sparsity		Speedup			Robo	VQA
Prefill	Decode	Prefill	Decode	E2E		
0%	0%	1.0×	1.0×	1.0×	86.4	
90%	0%	14.6 ×	1.1×	1.4 ×	80.0	
70%	85%	4.9×	1.2 ×	1.4 ×	89.1	

Table 1 | Decoupled Prefill–Decode Sparsity

Analysis. Retrieved tokens in SparseVILA exhibit two distinct functional roles: *Visual Attention Sinks* and *Visual Retrieval Tokens*. Sink tokens maintain stable activation across queries, acting as persistent attractors that stabilize cross-modal attention, whereas retrieval tokens vary dynamically with query content, capturing task-specific relevance. This separation explains how SparseVILA sustains contextual grounding while enabling efficient, query-adaptive retrieval. Qualitative analyses in Section 4.6 corroborate these behaviors, aligning with observations from VisionZip [13] and Visual Attention Redistribution (VAR) [32].

4. Experiments

4.1. Setup

Multi-Turn Evaluation. Many existing benchmarks generate multiple question–answer pairs for each image: *e.g.*, up to 18 in POPE [25]. However, evaluation protocols often remain confined to single-turn settings. In this work, we make use of the inherent multi-turn structure of such datasets to enable more efficient and accurate assessment of VLMs. We organize questions associated with the same image into coherent multi-turn conversations, which allows visual tokens to be prefilled only once. This setup not only reduces computational overhead but also better reflects realistic VLM usage in interactive settings. However, a potential problem is information leakage between turns, where earlier questions can unintentionally reveal answers to later ones. For example, in the GQA dataset [14]:

- Q1: “What is the person in front of *the sky* doing?”
- Q2: “What’s the person in front of?”

Here, Q1 discloses information that effectively answers Q2, enabling the model to respond correctly even without relying on the image. To mitigate this, we implement a partial KV cache eviction strategy: after each round, we remove only the KV entries corresponding to the previous question and answer. This preserves the efficiency gains of visual KV cache reuse while preventing unintended context carryover between turns. We will release our multi-turn evaluation framework to support the broader multimodal research community.

Baselines. We compare our SparseVILA with two categories of token pruning baselines: (i) *query-agnostic methods*, which prune redundant visual tokens without reference to the textual input, including VisionZip [13], PruMerge [24], and HIRED [26]; and (ii) *query-aware methods*, which adapt token selection based on the

	Sparsity		Speedup			AI2D	Chart QA	Doc VQA	GQA	Info VQA	MME Sum	POPE	SQA	Text VQA	
	P	D	E	P	D										
LLaVA-NeXT-7B	0	0	1.0×	1.0×	1.0×	63.9	53.0	63.6	63.5	28.4	1857.8	84.5	69.3	58.2	
+ FastV	.80	0	1.0×	1.5×	1.2×	1.2×	61.8	31.6	33.5	55.3	22.0	1568.2	76.7	66.7	52.7
+ SparseVLM	.75	0	1.0×	1.4×	1.2×	1.2×	63.2	39.9	41.8	59.7	22.2	1823.9	83.4	69.6	57.6
+ PDrop	.64	0	1.0×	1.3×	1.2×	1.2×	62.9	33.6	25.3	54.6	20.4	1793.4	81.6	68.9	52.6
+ PruMerge	.80	0	0.3×	1.5×	1.2×	1.2×	60.0	25.4	26.9	59.8	21.4	1686.2	82.0	67.8	48.3
+ HIRED	.80	0	0.9×	1.5×	1.2×	1.2×	59.4	33.4	34.8	60.4	22.0	1560.0	80.5	68.7	50.6
+ VisionZip	.80	0	1.0×	1.5×	1.2×	1.2×	62.9	38.2	48.5	60.3	24.2	1727.4	84.1	67.9	57.1
+ SparseVILA	.60	.75	1.0×	1.1×	1.2×	1.2×	64.1	47.8	58.0	62.7	25.6	1831.0	85.8	69.6	59.1
InternVL3.5-8B	0	0	1.0×	1.0×	1.0×	1.0×	80.9	79.2	86.4	60.5	72.8	2309.4	88.1	96.8	75.3
+ FastV	.80	0	1.0×	2.6×	1.4×	1.6×	67.8	28.4	23.7	48.3	26.9	1909.4	72.5	80.5	54.8
+ SparseVILA	.45	.95	1.0×	1.5×	1.6×	1.6×	77.0	61.8	56.4	58.9	58.1	2276.3	87.9	94.3	67.5
Nemotron-Nano-VL-8B	0	0	1.0×	1.0×	1.0×	1.0×	82.2	86.3	90.6	64.6	76.1	1936.4	87.7	97.2	82.9
+ SparseVILA	.60	.75	1.0×	1.1×	1.3×	1.3×	79.7	66.1	71.7	63.6	60.1	1859.8	87.4	96.2	71.2

Table 2 | **Image Benchmark Results.** SparseVILA preserves near-lossless accuracy on general VQA tasks and reduces degradation on document and chart benchmarks by a large margin compared to prior pruning methods, demonstrating stronger retention of fine-grained visual details.

language context, such as FastV [12], PDrop [33], and SparseVLM [22]. Most of these approaches estimate token salience using attention weights, which can be memory- and latency-intensive in visual encoders, often exceeding GPU limits at long context lengths. To ensure a fair comparison, we compute attention maps for these methods in a chunked manner to fit within memory constraints. In contrast, SparseVILA employs our fused Triton kernel to avoid materializing the full attention map (Figure 5).

Inference Setting. We build an optimized inference pipeline based on TinyChat. Specifically, we apply W8A8 quantization to the visual encoder following SmoothQuant [34], and W4A16 quantization to the LLM following AWQ [35]. This quantized version achieves a $2.4\times$ end-to-end speedup over the vanilla one, with negligible accuracy degradation, as verified in preliminary experiments. All subsequent results in this work are reported on top of this quantized version. Unless otherwise stated, inference is performed on a single NVIDIA A6000 GPU using greedy decoding with a batch size of 1.

Latency Evaluation. We measure the end-to-end inference runtime, including the visual encoder (E), language model prefilling (P), and decoding throughput (D). Total latency (E2E) is defined as the sum of prefill time and per-token decoding time, with decoding lengths fixed to ensure consistency across tasks. Because our evaluation focuses on multi-turn conversations, we account for the chunked prefilling cost of queries across conversation rounds and amortize it into the initial image prefill stage. The number of rounds for each task is set to the average number of conversational turns observed in its dataset. For image-based tasks, we fix the decoding length to 50 tokens per round to emulate image captioning workloads. For video-based tasks, we use 250 tokens per round to approximate the latency of video captioning and detailed video generation. Reasoning models are evaluated in a single-turn setting, where total latency is computed as the sum of prefill and decoding times over 1,500 tokens, consistent with the typical 1-2K token output length of reasoning tasks.

Sparsity Ratio. Our sparsity configuration adopts a straightforward approach for both prefill and decoding, ensuring efficient implementation and cross-model compatibility. Specifically, we set a constant prefill sparsity before the LLM and a uniform decoding sparsity across all layers. More granular strategies, such as layer-wise or head-aware sparsity, may yield further optimization but introduce additional complexity and tuning overhead. We prioritize simplicity and generalization, leaving these refinements for future work.

4.2. Image Benchmark Results

We evaluate SparseVILA across nine vision-language benchmarks, including AI2D [36], ChartQA [37], DocVQA [38], GQA [14], InfoVQA [39], MME [40], POPE [25], ScienceQA [41], and TextVQA [42]. These benchmarks span diagram understanding, document reasoning, and general visual question answering, enabling a comprehensive evaluation of efficiency-accuracy trade-offs.

As shown in Table 2, when applied to LLaVA-NeXT-7B [43], SparseVILA maintains near-lossless performance on general VQA tasks such as AI2D, GQA, POPE, and ScienceQA, achieving accuracy comparable to or even higher than the unpruned baseline under high sparsity. On fine-grained document and chart understanding benchmarks (ChartQA, DocVQA, and InfoVQA), SparseVILA exhibits over **15% less degradation** than prior pruning and merging methods such as FastV, SparseVLM, and VisionZip. These improvements stem from the decoupled sparsity design, which balances lightweight prefill pruning with adaptive decoding retrieval, preserving essential visual context across diverse vision-language tasks.

In addition to LLaVA-NeXT-7B, we also evaluate SparseVILA on InternVL3.5-8B [44] and Llama-Nemotron-Nano-VL-8B [45]. Across both models, SparseVILA maintains competitive accuracy while accelerating inference, demonstrating the generality of our decoupled sparsity framework across architectures and tasks.

4.3. Video Benchmark Results

We then evaluate SparseVILA on diverse video benchmarks covering video question answering, captioning, and retrieval. These tasks assess the model’s ability to handle extended visual contexts, sustain coherent generation over long sequences, and preserve fine-grained visual memory across multi-turn interactions. Across all benchmarks, SparseVILA delivers consistent improvements in both efficiency and accuracy by decoupling sparsity between the prefill and decoding stages. By shifting sparsity toward decoding, where query-aware retrieval selects only the most relevant visual tokens from the cached context, SparseVILA achieves faster throughput and stronger long-context retention than prior pruning-based methods, while maintaining fidelity in temporal understanding and generation quality.

4.3.1. Video Understanding

We evaluate SparseVILA on four long-context video understanding benchmarks: LongVideoBench [46], MLVU [47], NExT-QA [48], and Video-MME [49]. As shown in Table 3, SparseVILA consistently outperforms baselines across models [50, 51, 45]. Query-aware methods such as FastV, SparseVLM, and PDrop fail to scale beyond 32 frames due to their reliance on full joint query–vision attention, while query-agnostic methods like VisionZip and PruMerge introduce substantial overhead from token clustering and merging, which can even slow inference despite token reduction. In contrast, SparseVILA’s decoupled sparsity framework scales efficiently to long video contexts by combining query-agnostic pruning during prefill with query-aware retrieval during decoding. This design achieves up to **6.0 \times** faster language model prefill, **2.5 \times** faster decoding, and an overall **2.6 \times** end-to-end speedup, while maintaining near-lossless accuracy across all video understanding benchmarks.

Unlike the image benchmarks, SparseVILA even improves accuracy over the unpruned baseline on video benchmarks. We attribute this to more precise token retrieval enabled by a compact and information-dense KV cache, which helps the model focus on the most relevant visual cues. This also aligns with findings from StreamingLLM [52], where smaller active contexts were shown to improve focus and reasoning. Overall, these results indicate that decoupling sparsity not only enhances efficiency but also sharpens the model’s attention to semantically important information, improving both accuracy and scalability in long-context video understanding.

4.3.2. Video Captioning

We further evaluate SparseVILA on long-generation tasks using the VideoChatGPT benchmark [53], which measures a model’s ability to produce extended, free-form video descriptions under long-context settings. This benchmark provides a GPT-aided evaluation across five dimensions: *correctness*, *detail*, *contextual understanding*, *temporal understanding*, and *consistency*.

As shown in Table 4, SparseVILA maintains high generation quality while achieving substantial com-

	Sparsity		Speedup			LVB	MLVU	NExT-QA	Video-MME (w/o sub)				
	P	D	E	P	D	E2E	val	m-avg	mc	S	M	L	Overall
LongVILA-7B (256f)	0	0	1.0×	1.0×	1.0×	1.0×	53.8	64.9	78.6	67.6	57.7	51.2	58.8
+ VisionZip	.95	0	0.9×	28.5×	1.5×	2.1×	47.0	60.4	75.5	58.0	51.6	47.0	52.2
+ PruMerge	.95	0	0.9×	28.5×	1.5×	2.1×	47.9	60.9	75.7	57.9	51.6	46.7	52.0
+ SparseVILA	.75	.90	1.0×	5.1×	1.6×	2.1×	54.1	65.3	79.0	68.3	58.2	49.6	58.7
Qwen2.5-VL-7B (4fps)	0	0	1.0×	1.0×	1.0×	1.0×	59.2	65.5	76.0	73.0	60.8	53.1	62.3
+ SparseVILA	.75	.90	0.4×	6.0×	2.0×	1.9×	60.1	70.7	81.9	75.9	65.9	57.1	66.3
Nemotron-Nano-VL-8B (256f)	0	0	1.0×	1.0×	1.0×	1.0×	55.3	60.9	75.8	68.3	51.6	45.8	55.2
+ SparseVILA	.75	.95	1.0×	4.0×	2.5×	2.6×	55.9	63.1	76.6	68.9	54.2	46.8	56.6

Table 3 | **Video Understanding Benchmark Results.** SparseVILA delivers up to **6.0×** faster language model prefill, **2.5×** faster decoding, and **2.6×** end-to-end speedup, maintaining **near-lossless** accuracy on video understanding.

	Sparsity		Speedup			Video-ChatGPT						
	P	D	E	P	D	E2E	CI	DO	CU	TU	C	Overall
LongVILA-7B (256f)	0	0	1.0×	1.0×	1.0×	1.0×	2.34	2.21	2.81	1.70	2.46	2.31
+ VisionZip	.95	0	0.9×	28.5×	1.5×	2.1×	2.04	2.03	2.56	1.71	2.11	2.09
+ PruMerge	.95	0	0.9×	28.5×	1.5×	2.1×	2.07	2.00	2.57	1.75	2.10	2.10
+ SparseVILA	.75	.90	1.0×	5.1×	1.6×	2.1×	2.35	2.27	2.85	1.90	2.39	2.35

Table 4 | **Video Captioning Benchmark Results.** SparseVILA delivers **5.1×** faster prefill, **1.6×** faster decoding and **2.1×** end-to-end speedup while slightly improving the overall Video-ChatGPT score. It delivers consistent gains across *correctness* (CI), *detail* (DO), *contextual understanding* (CU), *temporal understanding* (TU), and *consistency* (C), outperforming baselines with more coherent video generation. Evaluation scores are obtained using `gpt-4o-mini-2024-07-18`.

putational savings. Compared to the unpruned baseline, it improves the overall Video-ChatGPT score from **2.31** to **2.35**, with a **0.2** gain in temporal understanding. Unlike VisionZip and PruMerge, which both suffer moderate degradation in contextual understanding, SparseVILA preserves coherence and factual grounding even under high decoding sparsity. It achieves up to a **2.1×** end-to-end speedup, demonstrating that query-aware retrieval during decoding effectively maintains semantic grounding and enhances detail richness, while directly addressing the dominant latency bottleneck in multimodal generation.

4.3.3. Visual Retrieval

We evaluate long-context retrieval performance on a multi-turn Visual Needle-in-a-Haystack (V-NIAH) benchmark, extended from LongVILA [50] and LongVA [54]. Each sequence contains five target “needles” interleaved among haystack frames. To emulate realistic conversational interaction, all needles are embedded within the haystack, and the model is prompted sequentially with one of the five corresponding queries, while the remaining needles serve as distractors. This design ensures that accurate retrieval depends on identifying the correct visual segment rather than leveraging correlations among other embedded needles. To account for depth sensitivity, we re-prefill the context at each depth by conditioning on the first query, reapplying pruning, and then issuing the target question for evaluation.

Using LongVILA-7B [50], we compare our SparseVILA with SparseVLM and FastV across progressively longer visual contexts. Both SparseVLM and FastV fail to scale beyond 32 frames due to their reliance on joint query–vision attention, which results in excessive memory consumption. Even within this range, they show early degradation in retrieval accuracy as context length increases. In contrast, SparseVILA sustains near-perfect retrieval up to 200 frames, demonstrating strong long-context retention (see Figure 6). This robustness stems from SparseVILA’s query-aware decoding sparsity, which selectively retrieves relevant visual cues from the preserved KV cache instead of pruning them during the prefill stage.



Figure 6 | **Visual Retrieval Results.** SparseVLM and FastV degrade and fail beyond 32 frames (8K context), while SparseVILA maintains perfect retrieval up to 200 frames, demonstrating superior long-context scalability.

4.4. Reasoning Benchmark Results

We evaluate SparseVILA on long-context and physical reasoning workloads that stress multimodal inference beyond standard VQA or captioning. These tasks typically require substantially longer generations, making decoding throughput the dominant contributor to end-to-end latency. Prior sparsity methods concentrate compression in the prefill stage and thus provide limited benefit in this regime. In contrast, SparseVILA’s decoupled design allocates lightweight, query-agnostic pruning to prefill while shifting aggressive, query-aware retrieval to decoding, preserving reasoning fidelity under long outputs and delivering practical speedups.

4.4.1. Video Reasoning

We assess SparseVILA on LongVideo-Reason [55], which features complex question–answer pairs requiring temporal reasoning over extended video sequences. As reported in Table 5, SparseVILA consistently outperforms state-of-the-art pruning and merging approaches (*e.g.*, PruMerge, VisionZip) while achieving up to **1.3 \times** faster inference. The gains stem from reallocating sparsity toward decoding, where query-aware retrieval narrows attention to the most relevant visual tokens in the cached context without discarding information needed for later turns. This maintains long-horizon temporal consistency and yields higher answer accuracy at comparable end-to-end speedups.

4.4.2. Physical Reasoning

We evaluate SparseVILA on physical reasoning suites that demand causal understanding and multi-step deduction. As shown in Table 6, SparseVILA matches or exceeds baselines across all tasks, operating on the Pareto frontier of efficiency and accuracy. Notably, SparseVILA surpasses the unpruned model on all subsets at 24 frames-per-second while delivering a lossless **1.9 \times** end-to-end speedup and a **4.5%** performance gain. These results indicate that concentrating sparsity in decoding sharpens the model’s focus on semantically critical evidence, preserving structured reasoning under aggressive compression.

Discussion. Across both video and physical reasoning settings, many pruning methods reduce theoretical compute yet incur substantial overhead from salience computation or token reorganization, limiting realized speedups. In contrast, SparseVILA combines stage-aware sparsity with fused Triton kernels (Figure 7), keeping overhead low and aligning empirical latency with theoretical gains. The decoupled design retains a rich visual KV cache for future turns while activating only the query-relevant subset during generation, yielding robust accuracy and consistent acceleration in reasoning-heavy workloads.

4.5. Efficiency Analysis

SparseVILA achieves consistent acceleration across image, video, and reasoning workloads through its decoupled sparsity framework. This design scales effectively across diverse architectures and attention mechanisms, including standard multi-head attention (MHA) in LLaVA-NeXT [43] and grouped-query attention (GQA) in LongVILA-7B [50], Qwen2.5VL-7B [56], and InternVL3.5-8B [44]. By combining custom kernel design with stage-aware sparsity allocation, SparseVILA provides a full-stack optimization of the VLM inference pipeline, spanning embedding, prefill, and decoding. This holistic design captures both compute- and memory-bound stages, improving end-to-end latency while maintaining fidelity. Furthermore, SparseVILA’s decoupled sparsity scales seamlessly from short-context image tasks to long-horizon video and reasoning

	Sparsity		Speedup			Temporal	Goal	Plot	Spatial	Overall
	P	D	E	P	D					
LongVILA-R1-7B (512f)	0	0	1.0×	1.0×	1.0×	1.0×	75.5	88.9	74.9	58.5
+ PruMerge	.95	0	0.8×	10.5×	1.1×	1.1×	63.3	86.5	73.9	57.3
+ VisionZip	.95	0	0.9×	10.5×	1.1×	1.2×	62.9	85.4	70.3	61.0
+ SparseVILA	.75	.90	1.0×	4.5×	1.2×	1.3×	66.7	87.8	73.1	59.8
										74.4

Table 5 | **Video Reasoning Benchmark Results.** SparseVILA maintains competitive performance on long-video reasoning tasks while delivering up to **1.3×** end-to-end speedup.

	Sparsity		Speedup			HoloAssist	RoboFail	RoboVQA	Average
	P	D	E	P	D				
Cosmos-Reason1-7B (4fps)	0	0	1.0×	1.0×	1.0×	1.0×	65.0	60.0	86.4
+ PruMerge	.90	0	0.2×	2.2×	1.1×	0.7×	41.0	39.0	52.3
+ VisionZip	.90	0	0.2×	14.6×	1.1×	0.8×	66.0	54.0	80.3
+ FastV	.71	0	1.0×	2.2×	1.1×	1.3×	46.0	37.0	80.9
+ SparseVILA	.70	.85	0.7×	4.9×	1.2×	1.4×	64.0	63.0	89.1
									72.0
Cosmos-Reason1-7B (24fps)	0	0	1.0×	1.0×	1.0×	1.0×	72.0	54.0	88.2
+ PruMerge	.97	0	0.04×	13.8×	1.1×	0.7×	46.0	43.0	70.0
+ VisionZip	.97	0	0.04×	73.4×	1.6×	0.3×	64.0	54.0	80.9
+ SparseVILA	.75	.95	0.4×	7.6×	2.0×	1.9×	75.0	58.0	94.5
									75.9

Table 6 | **Physical Reasoning Benchmark Results.** SparseVILA delivers up to **7.6×** faster language model prefill, **2.0×** faster decoding, and **1.9×** end-to-end speedup, while outperforming prior methods and the baseline model at 24 frames-per-second.

settings, ensuring stable efficiency across heterogeneous model families. Besides, SparseVILA’s prefill-stage pruning provides complementary memory savings, reducing KV memory usage by **72.5%** and linear FLOPS by **87.6%** on LongVILA-7B through structured token sparsity.

Decoding Attention Kernel Efficiency. We further analyze performance at the kernel level to isolate the contribution of our optimization from model-level sparsity. The decoding stage of LLM inference is memory-bound and thus bottlenecked by memory movement. By reducing the effective size of the KV Cache, SparseVILA lowers both memory traffic and decoding FLOPs, leading to substantial acceleration. As shown in Figure 7, SparseVILA delivers up to **11.4×** speedup on long-context video workloads and **6.8×** on reasoning tasks.

Empirical vs. Theoretical Latency Analysis A key factor in evaluating sparsity strategies is the gap between theoretical and realized latency. Even when token reduction establishes a clear upper bound on achievable speedup, additional computation can diminish these gains in practice. The reported latency measurements therefore capture both the benefits of sparsity and the method-specific overhead incurred during inference. This overhead arises from pruning metric computation, token reorganization, and selection logic. Query-aware methods, which delay pruning to deeper layers, introduce nontrivial computational cost. VisionZip [13] exhibits significant overhead in the embedding stage due to full attention weight computation, limiting effective speedup at long contexts. Similarly, PruMerge [24] incurs additional prefill-stage overhead due to clustering-based pruning. In contrast, SparseVILA maintains low overhead in both prefill and decoding, resulting in empirical latency that more closely aligns with the theoretical sparsity bound. Table 7 summarizes the measured CUDA-time overhead for select methods across video and reasoning workloads.

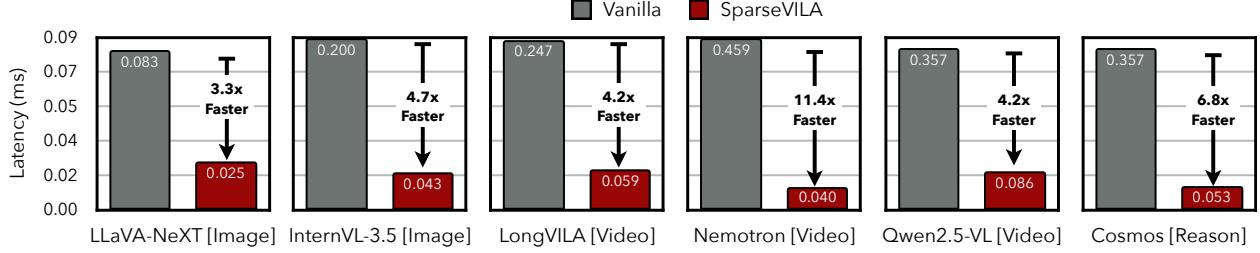


Figure 7 | **Decoding Attention Kernel.** SparseVILA’s decoupled sparsity reduces the size of the effective KV cache in decoding, delivering up to **11.4×** speedup on the attention kernel for long video understanding tasks.

Method	LongVILA-7B (256f)				Cosmos-Reason1-7B (4fps)			
	Overhead		Speedup		Overhead		Speedup	
	CUDA (ms)	%	T	E	CUDA (ms)	%	T	E
PruMerge	448.3	1.84	7.41×	6.49×	20990.1	171.4	3.04×	0.49×
VisionZip	206.3	0.85	7.41×	6.94×	17612.3	143.8	3.04×	0.57×
SparseVILA	94.9	0.39	3.98×	3.91×	1400.9	26.1	2.28×	1.81×

Table 7 | **Overhead comparison across workloads.** Comparison of the overhead incurred by different methods on LongVILA-7B (video), and Cosmos-Reason1-7B (reasoning) workloads. CUDA Time (ms) denotes the additional latency measured on a single NVIDIA A6000 GPU, and the percentage indicates the relative overhead during the media-prefilling stage. We additionally report the theoretical (T) and empirical (E) speedups for each model/workload corresponding to settings in Section 4.

4.6. Qualitative Analysis

Emergence of sink and retrieval tokens. In Figure 8, we examine how visual *sink* and *retrieval* tokens emerge throughout the LLM layers of LLaVA-1.5. Profiling the attention maps reveals that early layers concentrate on a small subset of visual tokens that remain stable across different queries – these correspond to persistent *visual sink tokens* that act as anchors of scene understanding. As depth increases, attention patterns diversify and *retrieval tokens* emerge, focusing selectively on regions relevant to the query. The sink tokens continue to exist but with diminished strength, indicating that query-specific reasoning gradually overrides the globally salient structures. Quantitatively, we can use the intersection-over-union (IoU) of selected tokens for different input queries to quantify the proportion of sink and retrieval tokens captured. On the first 38 multi-turn queries in the GQA dataset, the IoU is highest in shallow layers (e.g., Layer 2), confirming strong sink consistency over different queries, and decreases toward deeper layers (e.g., Layer 19), where query-dependent retrieval dominates.

Design implications for SparseVILA. Considering that retrieval tokens only emerge in deeper layers, pruning tokens early in the network – as done by many prefill-only methods – irreversibly removes information essential for query-aware reasoning. SparseVILA, therefore, is specifically designed to preserve maximum content in the prefill phase, thereby deferring the bulk of pruning decisions into the decoding stage. It then performs selection across all layers, ensuring that both persistent sinks and query-dependent retrievals are retained. By shifting aggressive sparsity into the decoding stage, SparseVILA exploits this separation to balance compression and contextual fidelity, preserving long-term grounding without redundant visual computation.

SparseVILA token selection. To visualize this effect, Figure 9 shows token selection frequencies under 50% context sparsity and 75% decoding sparsity. Each heatmap depicts how often a visual token is chosen across all LLM layers for a given query. SparseVILA consistently preserves sink tokens – regions repeatedly selected across layers – while dynamically retrieving query-specific tokens corresponding to objects of interest. As illustrated in the examples, retrieval focuses shift appropriately across questions (e.g., the cyclist vs.

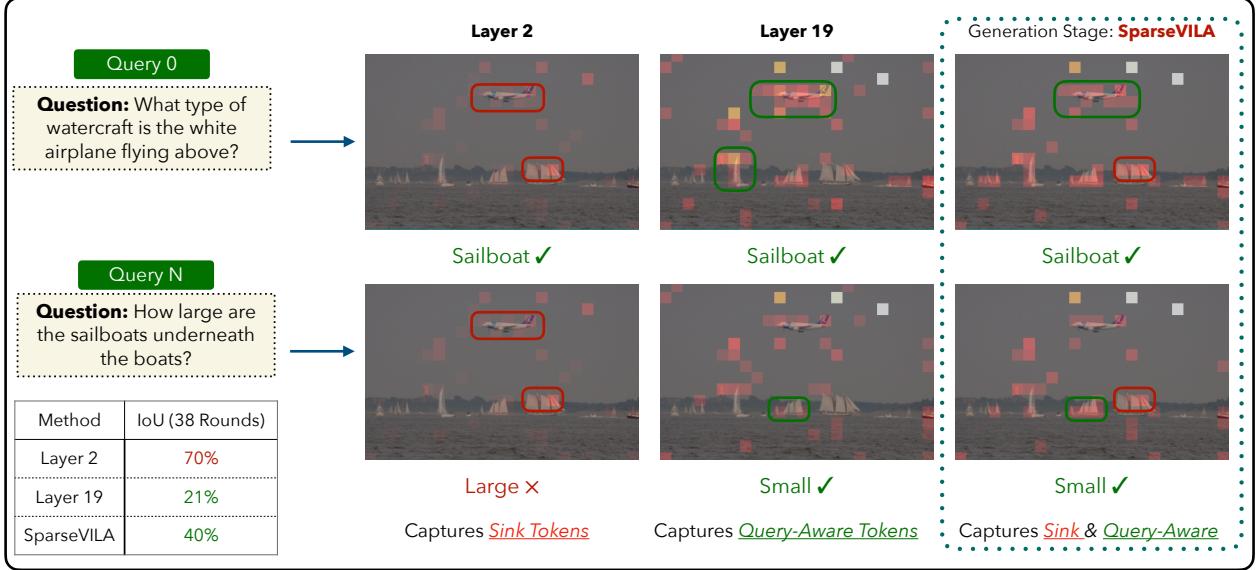


Figure 8 | **Emergence of sink and retrieval tokens.** Early layers emphasize persistent visual sinks, while deeper layers highlight query-dependent retrieval tokens. This observation motivates SparseVILA’s decoupled design, which preserves both phenomena through full-context attention before sparsification.

the signpost), demonstrating that the decoupled design maintains both stability and adaptability in token selection.

5. Related Work

5.1. Visual Token Compression

Early studies on visual token compression focus on vision transformers (ViTs), where token pruning [57, 58, 16, 59, 60, 61], token merging [15, 17], and compact token representation learning [62] improve throughput by reducing redundant computations. Building on these ideas, many recent methods extend token compression to vision-language models (VLMs). Approaches such as LLaVA-PruMerge [24], HIRED [26], and VisionZip [13] selectively prune redundant visual tokens after the encoder using attention-based salience metrics. However, their query-agnostic nature leads to significant degradation under high sparsity.

To address these limitations, query-aware pruning methods emerge. FastV [12] uses early LLM attention maps to guide token selection based on the query. SparseVLM [22] uses cross-modal attention scores to remove visually irrelevant tokens during prefilling. While these methods preserve accuracy for single-turn queries, they struggle in multi-turn conversations: once a token is pruned, it cannot be recovered for future queries, leading to cumulative information loss. As shown in Figure 2, query-aware pruning exhibits rapid degradation over consecutive rounds, often underperforming query-agnostic methods when the context persists across turns.

This trade-off motivates SparseVILA, which decouples sparsity across the inference pipeline. During the prefill stage, SparseVILA performs query-agnostic pruning to remove redundant or uninformative visual tokens while retaining a comprehensive visual cache. During decoding, it retrieves only query-relevant tokens from the preserved cache, achieving significant end-to-end acceleration without compromising multi-turn consistency or contextual fidelity. By distributing sparsity between stages, SparseVILA combines the generalization of query-agnostic pruning with the adaptivity of query-aware retrieval.

5.2. KV Cache Compression

In long-context LLM and VLM inference, the key-value (KV) cache grows linearly with sequence length, imposing substantial latency and memory overhead. Recent work introduces a variety of KV cache compression techniques to address this challenge. StreamingLLM [52] maintains a finite cache by preserving attention

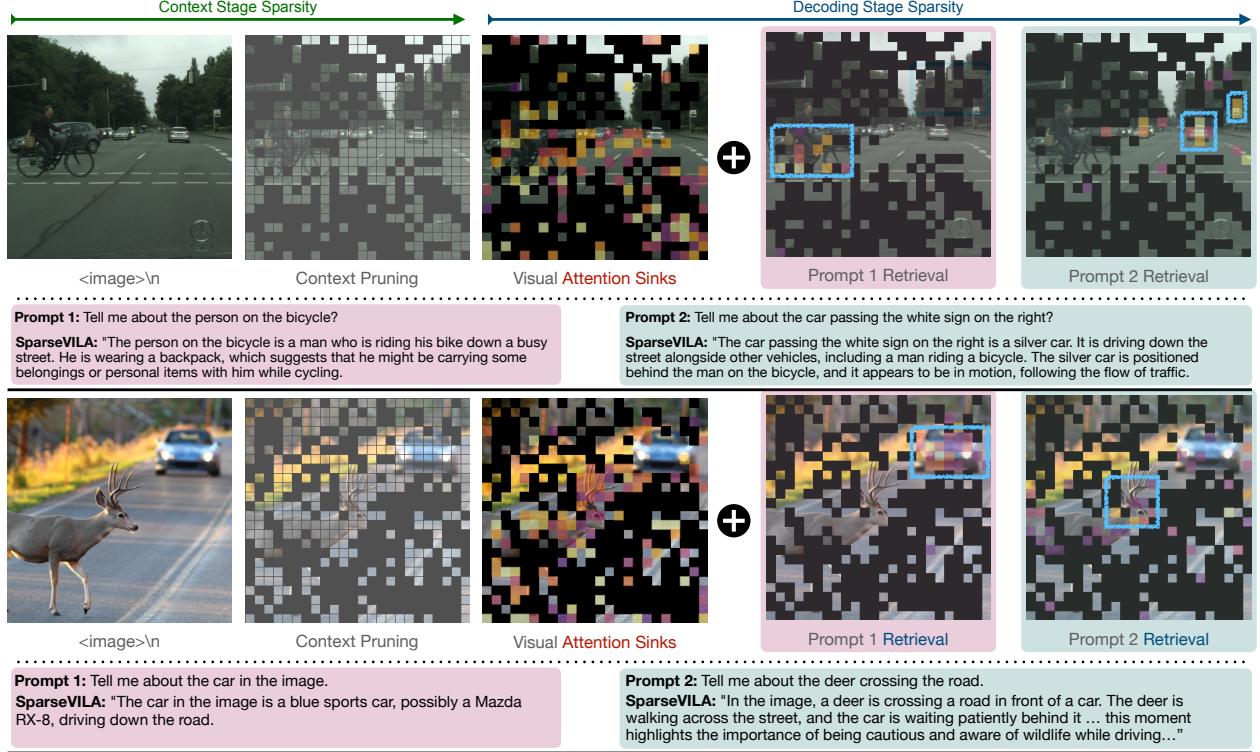


Figure 9 | **Token selection frequency under 50% context and 75% decoding sparsity.** Each heatmap shows how often a visual token is selected across all LLM layers for a given query on LLaVA-1.5. SparseVILA maintains stable sink tokens while retrieving query-specific tokens, achieving efficient yet faithful decoding.

sinks and discarding older context tokens. SnapKV [63] predicts token importance within an observation window to avoid redundant storage. H2O [64] identifies essential KV entries based on cumulative historical attention scores, prioritizing the most influential tokens. Although very effective for text-based applications, these methods often discard cache entries that remain relevant for future decoding steps, which limits their reliability in multi-turn or temporally extended reasoning.

Quest [65] estimates upper-bound attention scores for each page to preserve critical tokens. LazyLLM [66] defers KV computation until the corresponding tokens are required. DuoAttention [67] separates retrieval and streaming heads, assigning full caches to retrieval heads and fixed-length caches to streaming heads. While these strategies efficiently reduce decoding latency for text-only LLMs, they overlook the structured spatial and temporal sparsity inherent to image and video inputs. Visual and video understanding tasks naturally exhibit redundancy across frames and regions, offering further potential for selective retention and retrieval.

SparseVILA complements these approaches by integrating visual token sparsity with query-aware KV cache retrieval. Instead of discarding visual context, it preserves a compact but reusable visual cache that supports dynamic retrieval across conversation rounds. This design enables efficient multimodal decoding, sustaining performance under long-context, multi-turn interaction while avoiding the information loss typical of purely text-based compression methods.

6. Conclusion

We present SparseVILA, a unified sparsity framework that accelerates Vision Language Model (VLM) inference by decoupling visual compression across prefill and decoding. SparseVILA prunes redundant visual tokens during prefill and selectively retrieves query-relevant tokens during decoding, reducing latency where it matters most. This design scales from short-context image tasks to long-horizon video and reasoning workloads, where decoding dominates overall inference time. Considering the entire VLM inference stack – visual embedding, prefill, and decoding – SparseVILA achieves up to **4.0 \times** faster prefilling, **2.5 \times** faster

decoding, and $2.6\times$ end-to-end speedup, while preserving or improving accuracy on multi-turn and reasoning benchmarks. Unlike prior pruning methods that trade speed for capability, SparseVILA maintains fidelity across modalities and architectures through decoupled sparsity allocation and efficient kernel design. This establishes a scalable, training-free foundation for accelerating the next generation of multimodal systems.

References

- [1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. [1](#) [3](#)
- [2] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. VILA: On Pre-training for Visual Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [1](#)
- [3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv:2308.12966*, 2023. [1](#)
- [4] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution. *arXiv:2409.12191*, 2024. [1](#)
- [5] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Vishwesh Nath, Jinyi Hu, Sifei Liu, Ranjay Krishna, Daguang Xu, Xiaolong Wang, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. NVILA: Efficient Frontier Visual Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. [1](#)
- [6] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A Simple and Effective Pruning Approach for Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. [2](#)
- [7] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the Structural Pruning of Large Language Models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023. [2](#)
- [8] Mingjian Zhu, Yehui Tang, and Kai Han. Vision Transformer Pruning. *arXiv:2104.08500*, 2021. [2](#)
- [9] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. QServe: W4A8KV4 Quantization and System Co-Design for Efficient LLM Serving. In *Proceedings of the Conference on Machine Learning and Systems (MLSys)*, 2025. [2](#)
- [10] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. [2](#)
- [11] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. [2](#)
- [12] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. [2](#), [3](#), [7](#), [13](#)
- [13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. [2](#), [4](#), [6](#), [11](#), [13](#)
- [14] Drew A Hudson and Christopher D Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#), [6](#), [8](#)
- [15] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. [2](#), [13](#)
- [16] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Mengshu Sun, Wei Niu, Xuan Shen, Geng Yuan, Bin Ren, Minghai Qin, Hao Tang, and Yanzhi Wang. SPViT: Enabling Faster Vision Transformers via Soft Token Pruning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. [2](#), [13](#)
- [17] Samir Khaki and Konstantinos N Plataniotis. The Need for Speed: Pruning Transformers with One Recipe. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. [2](#), [13](#)

- [18] Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. SparseViT: Revisiting Activation Sparsity for Efficient High-Resolution Vision Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2
- [19] Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2
- [20] Daniel Bolya and Judy Hoffman. Token Merging for Fast Stable Diffusion. *arXiv:2303.17604*, 2023. 2
- [21] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned Token Pruning for Transformers. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022. 2
- [22] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 2, 3, 7, 13
- [23] Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. IVTP: Instruction-Guided Visual Token Pruning for Large Vision-Language Models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2
- [24] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 4, 6, 11, 13
- [25] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating Object Hallucination in Large Vision-Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. 3, 6, 8
- [26] Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. HiRED: Attention-Guided Token Dropping for Efficient Inference of High-Resolution Vision-Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. 4, 6, 13
- [27] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative Vision Foundation Model – Reduce All Domains Into One. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4
- [28] Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved Baselines for Agglomerative Vision Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 4
- [29] Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: An Intermediate Language and Compiler for Tiled Neural Network Computations. In *Proceedings of the ACM SIGPLAN International Workshop on Machine Learning and Programming Languages (MAPL)*, 2019. 4
- [30] Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 5
- [31] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mirchandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak Shafran, Brian Ichter, and Yuan Cao. RoboVQA: Multimodal Long-Horizon Reasoning for Robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 6
- [32] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See What You Are Told: Visual Attention Sink in Large Multimodal Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 6
- [33] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating Your Large Vision-Language Models via Pyramid Visual Redundancy Reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 7
- [34] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. 7
- [35] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-Aware Weight Quantization for LLM Compression and Acceleration. In *Proceedings of the Conference on Machine Learning and Systems (MLSys)*, 2024. 7

- [36] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A Diagram Is Worth A Dozen Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 8
- [37] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. 8
- [38] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A Dataset for VQA on Document Images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 8
- [39] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022. 8
- [40] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. *arXiv:2306.13394*, 2023. 8
- [41] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 8
- [42] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA Models That Can Read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8
- [43] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge. <https://llava-vl.github.io/blog/2024-01-30-llava-next/>, 2024. 8, 10
- [44] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Binqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv:2508.18265*, 2025. 8, 10
- [45] NVIDIA. Llama Nemotron Nano VL. <https://build.nvidia.com/nvidia/llama-3.1-nemotron-nano-vl-8b-v1>, 2025. 8
- [46] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A Benchmark for Long-Context Interleaved Video-Language Understanding. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 8
- [47] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking Multi-Task Long Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 8
- [48] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8
- [49] Chaoyou Fu, Yuhang Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-Modal LLMs in Video Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 8
- [50] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. LongVILA: Scaling Long-Context Visual Language Models for Long Videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 8, 9, 10
- [51] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv:2502.13923*, 2025. 8

- [52] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 8, 13
- [53] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. 8
- [54] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long Context Transfer from Language to Vision. *arXiv:2406.16852*, 2024. 9
- [55] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. Scaling RL to Long Videos. *arXiv:2507.07966*, 2025. 10
- [56] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 Technical Report. *arXiv:2407.10671*, 2024. 10
- [57] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which Tokens to Use? Investigating Token Reduction in Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 13
- [58] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 13
- [59] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch Slimming for Efficient Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 13
- [60] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 13
- [61] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 13
- [62] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint Token Pruning and Squeezing Towards More Aggressive Compression of Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 13
- [63] 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. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 14
- [64] 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. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 14
- [65] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 14
- [66] Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. LazyLLM: Dynamic Token Pruning for Efficient Long Context LLM Inference. *arXiv:2407.14057*, 2024. 14
- [67] Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 14