MMINFERENCE: ACCELERATING PRE-FILLING FOR LONG-CONTEXT VLMS VIA MODALITY-AWARE PERMUTATION SPARSE ATTENTION

Yucheng Li[◊], **Huiqiang Jiang**[†], **Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, Lili Qiu**
Microsoft Corporation, [◊]University of Surrey
yucheng.li@surrey.ac.uk,{hjiang,yuqyang}@microsoft.com
https://aka.ms/MMInference

ABSTRACT

The integration of long-context capabilities with visual understanding unlocks unprecedented potential for Vision Language Models (VLMs). However, the quadratic attention complexity during the pre-filling phase remains a significant obstacle to real-world deployment. To overcome this limitation, we introduce **MMInference** (*Multi-modality Million tokens Inference*), a dynamic sparse attention method that accelerates the pre-filling stage for long-context multi-modal inputs. First, our analysis reveals that the temporal and spatial locality of video input leads to a unique sparse pattern, the *Grid pattern*. Simultaneously, VLMs exhibit markedly different sparse distributions across different modalities. We introduce a permutation-based method to leverage the unique Grid pattern and handle modality boundary issues. By offline search the optimal sparse patterns for each head, MMInference constructs the sparse distribution dynamically based on the input. We also provide optimized GPU kernels for efficient sparse computations. Notably, MMInference integrates seamlessly into existing VLM pipelines without any model modifications or fine-tuning. Experiments on multi-modal benchmarks—including Video QA, Captioning, Vision-NIAH, and Mixed-Modality-NIAH—with state-of-the-art long-context VLMs (LongVila, Llava-Video, VideoChat-Flash, Qwen2.5-VL) show that MMInference accelerates the pre-filling stage by up to $8.3\times$ at 1M tokens while maintaining accuracy.

1 INTRODUCTION

Scaling the context size of Vision Language Models (VLMs) allows them to handle extended temporal information from long video and text inputs, which is crucial for various applications including robotics Black et al. (2024); Prasad et al. (2024); Cheang et al. (2024), autonomous driving Hu et al. (2023); Wang et al. (2024b); Gao et al. (2024), and healthcare Liu et al. (2024b). In addition, Zhang et al. (2024b) and Chen et al. (2025) show that scaling the context size of VLMs can improve the resolution in the temporal dimension and lead to better performance in video understanding tasks.

However, due to the quadratic complexity of attention, processing long multi-modal inputs (i.e.,

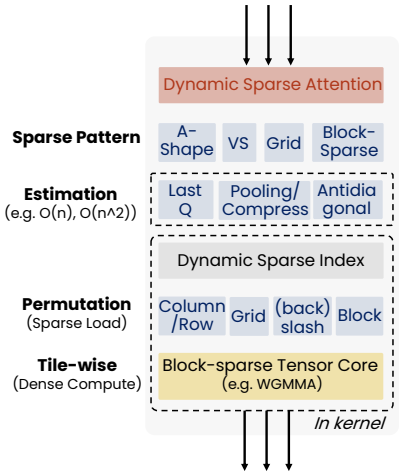
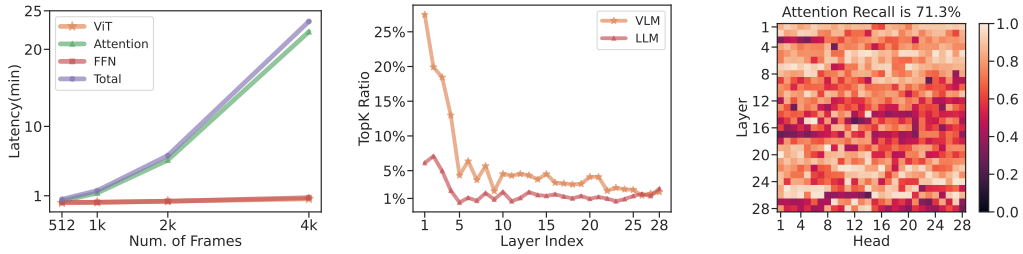


Figure 1: Dynamic sparse attention pipelines leverage sparse loading with dense computation (Zheng et al., 2023) to enable hardware-efficient acceleration.

[◊]Work during internship at Microsoft. [†]Corresponding author.



(a) VLMs' attention incurs heavy cost.

(b) VLMs' attention is sparse.

(c) Sparsity of VLMs' attention is dynamic.

Figure 2: (a) Latency breakdown of the pre-filling stage, with 256 tokens per frame. (b) How much element in attention needs to be computed to achieve 95% recall in a 128k context. (c) Low attention recall when reusing the top-k indices from a different request. Visualizations are based on LongVILA-7B-1M Chen et al. (2025) with a single A100.

the pre-fill stage) can take minutes prior to auto-
regressive decoding. As shown in Fig. 2a, this leads to significant Time-to-First-Token latency, which hinders the wide adoption of long-context VLMs in real-world applications. Previous work Child et al. (2019); Liu et al. (2022; 2024a); Yuan et al. (2025); Lu et al. (2025) reveals that attention matrices are typically sparse, prompting the development of sparse attention methods such as Sparse Transformer Child et al. (2019), Swin Transformer Liu et al. (2021), and StreamingLLM Xiao et al. (2024). More recently, MInference Jiang et al. (2024) proposes to use dynamic sparse attention that estimates the sparse index online, and leverages optimized GPU kernels for end-to-end acceleration. However, these methods fail to exploit the unique sparse patterns in long-context VLMs, and struggle with mixed or interleaved modalities, limiting their applicability without compromising performance.

Unlike long-text contexts, video and image inputs in VLMs exhibit spatiotemporal locality, forming grid-like attention patterns with evenly spaced vertical and horizontal lines (Fig. 3a). In mixed-modality inputs, clear modality boundaries emerge: attention across modalities diverges significantly from intra-modality attention (Fig. 3b). These factors pose unique challenges for exploiting sparsity to accelerate the pre-fill stage.

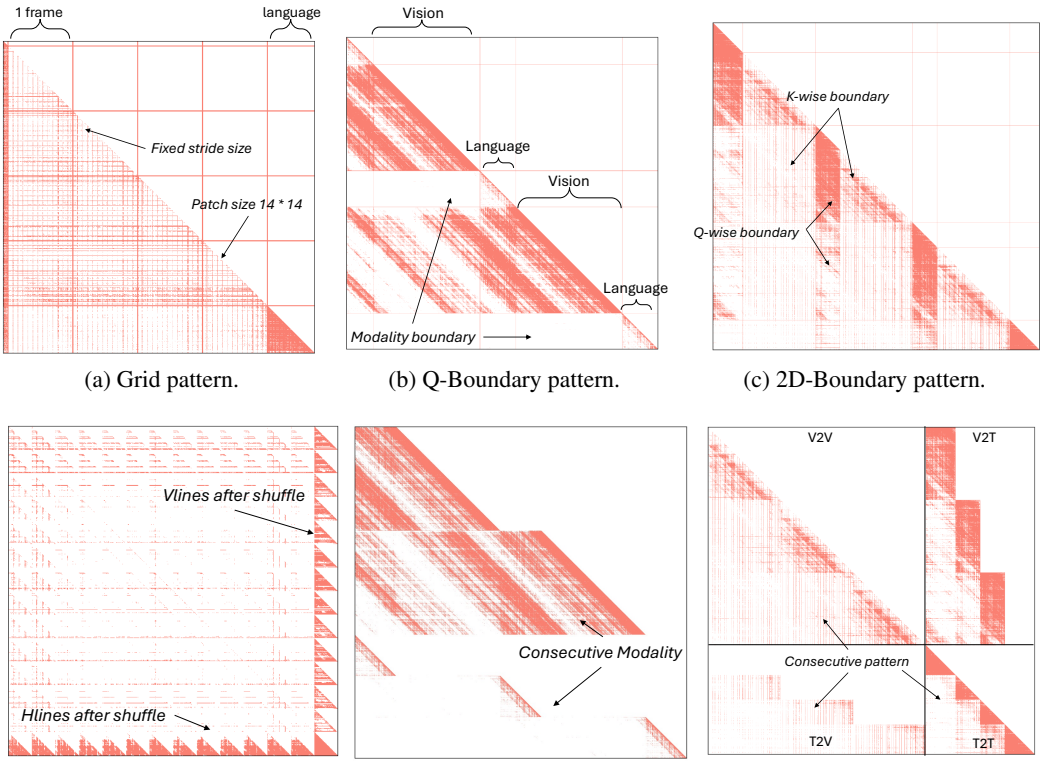
In this paper, we present MMInference, a permutation-based dynamic sparse attention method that significantly reduces attention FLOPs, accelerating the pre-fill stage of long-context VLMs. First, MMInference identifies the grid heads and leverages a *row- and column-wise permutation* to gather the sparse grid for efficient hardware computation. Next, we detect Query-boundary and 2D-boundary patterns to address inter-modality boundary issues, and apply a *modality-wise permutation* to isolate intra-modality regions. This results in a consecutive sparse index within each modality, permitting efficient hardware implementation of sparse computing. Finally, a *Modality-Aware Sparse Attention Search Algorithm* is devised to fine-tune both inter- and intra-modality patterns offline, to optimize performance with minimal overhead.

We conduct extensive experiments using four state-of-the-art long-context VLMs, Llava-Video Zhang et al. (2024b), LongVila Chen et al. (2025), VideoChat-Flash Li et al. (2025a) and Qwen2.5-VL Bai et al. (2025), across diverse video understanding tasks such as video captioning Maaz et al. (2024), video question answering Yu et al. (2019); Xiao et al. (2021); Mangalam et al. (2023); Fu et al. (2024), and video information retrieval Zhang et al. (2024a). Additionally, we propose the Mixed-Modality Needle in a Hackathon task to assess multi-modal input performance. Our method effectively addresses modality boundaries, significantly accelerates the prefilling stage, and maintains high accuracy. With a 1M-length context, it achieves speedups of up to 8.3× and 1.7× over FlashAttention-2 and MInference, respectively.

2 Attention Heads in VLMs

The sparsity of the attention operation in pre-trained text-only LLMs, particularly in long-context scenarios, has been extensively studied Wu et al. (2025); Ribar et al. (2024); Jiang et al. (2024); Li

et al. (2024), showing that only 3% of attention weights are activated while achieving a recall rate of 96.8%. Similarly, VLMs also demonstrate notable dynamic sparsity in long-context scenarios. This section examines the shared and distinct properties of text-only and multi-modal LLMs in long-context scenarios, focusing on attention sparsity, sparse patterns, and modality boundaries.



(a) Grid pattern. (b) Q-Boundary pattern. (c) 2D-Boundary pattern.

(d) Permuted Grid pattern. (e) Permuted Q-Boundary pattern. (f) Permuted 2D-Boundary pattern.

Figure 3: Visualization of pre- vs. post-permutation sparsity attention patterns in VLMs.

2.1 Multi-modality Attention is Dynamically Sparse

As illustrated in Fig. 2a, for a 128k × 128k attention matrix in VLMs, retaining only the top 5.78% of attention weights on average suffices to recall 95% of total attention, indicating that each token attends only to a limited subset of tokens, even in long sequences. However, VLMs exhibit lower sparsity than text-only LLMs, where only 1.79% of weights achieve a 95% recall rate. Notably, the bottom layers in VLMs (e.g., the first four layers in LongVila) show reduced sparsity. Yet, due to variability across attention heads, 52.3% of heads in VLMs require less than 2% of attention to be recalled. This highlights substantial computational redundancy in VLMs, especially in long-context scenarios.

Similarly to LLMs, while the sparse nature of attention matrices remains consistent across inputs, the specific distributions of sparse attention are highly dynamic. As shown in Fig. 2c, reusing top-k indices for 95% attention recall (derived from Fig. 2b) across different contexts leads to a significant drop in performance.

2.2 The Grid Head in VLMs

In long-context language modeling, efficient attention mechanisms like sliding window attention Jiang et al. (2023) and StreamingLLM Xiao et al. (2024) exploit the locality property of text sequences. However, multi-modal inputs introduce unique geometric structures that redefine locality. As shown in Child et al. (2019), image patches exhibit locality along both vertical and horizontal directions, forming local window and slash-like patterns. Similarly, video inputs maintain locality

across temporal and spatial dimensions, with frame-based sampling yielding more regular and predictable patterns.

We observe that certain VLM attention heads exhibit a **grid pattern**. While the grid's stride and starting position vary with context, the horizontal and vertical lines are evenly spaced and often symmetrical—a distinct behavior compared to text-only LLMs Jiang et al. (2024); Lai et al. (2025). Fig. 3a visualizes a grid head, demonstrating how local tokens in temporal and spatial dimensions are evenly distributed within the attention map, with attention focused primarily on these local tokens.

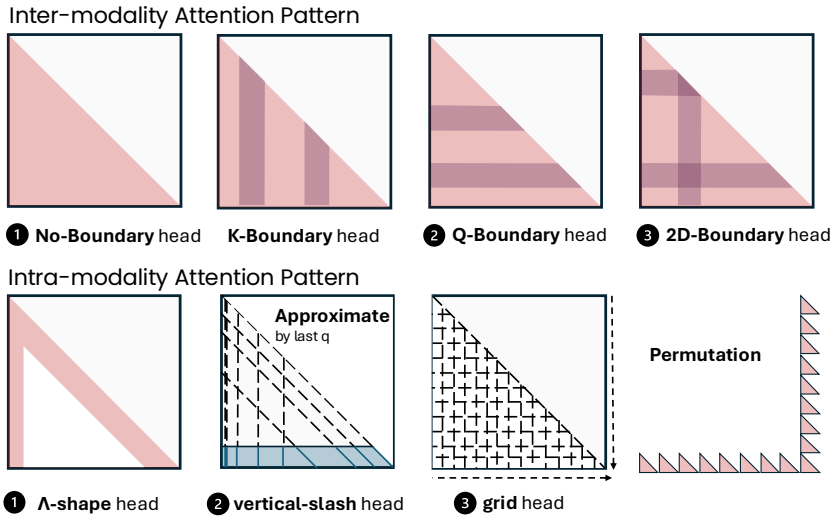


Figure 4: The framework of MMInference, encompassing both inter- and intra-modality sparse attention patterns.

2.3 Modality Boundaries in Multi-Modal Input

The input format of VLMs differs significantly from text-only LLMs. A dedicated vision encoder generates visual representations, which are processed alongside text embeddings by the LLM. Despite pretraining on large-scale datasets, the interactions and processing patterns between modalities vary considerably, leading to distinct modality boundaries in attention Tu et al. (2025), as illustrated in Fig. 3b and 3c.

Specifically, we observe two key characteristics: 1) *Intra-modality consistency*: Attention within each modality follows a consistent pattern. For instance, the vision region in Fig. 3b exhibits a clear slash pattern, where critical elements are effectively clustered. 2) *Modality-separated continuity*: Patterns within a modality can be interrupted by boundaries from other modalities. As shown in Fig. 3b, vision slashes are segmented by the boundary introduced by the language region.

We categorize the modality boundary patterns of VLMs into four distinct types: No-Boundary, K-Boundary, Q-Boundary, and 2D-Boundary, as illustrated in Figs. 3 and 4. 1) **No Boundary** and **K-Boundary** exhibit either no clear modality boundary or a boundary only along the key dimension, as shown in Fig. 6. Since continuity is maintained along the query dimension, these heads can be efficiently handled using intra-modality sparse patterns. 2) **Q-Boundary** refers to attention modality boundaries across the query dimension. For example, in Fig. 3b, sparse patterns like Text-to-Video and Video-to-Video appear interconnected, forming a trapezoidal structure, while a clear boundary separates Visual-to-Text and Text-to-Visual attention. 3) **2D-Boundary** occurs when modality boundaries are present in both query and key dimensions. As shown in Fig. 3c, the 2D modality boundary segments attention weights into distinct blocks. Additionally, our analysis of Audio LMs Chu et al. (2024) and end-to-end multimodal LMs Xu et al. (2025a); Li et al. (2025b) reveals that the cross-modality boundary phenomenon persists across these architectures. These boundaries pose unique challenges and hinder direct application of existing sparse attention methods to multi-modal inputs.

2.4 Sparse Distributions Continuity Across Boundaries

Although sparsity patterns in VLMs are often discontinuous across modalities due to modality boundaries, we find that sparsity distributions can remain continuous across these boundaries and extrapolate to other regions of the same modality. For example, in Fig. 3b, the slash lines maintain the same relative position across different areas of the vision modality. In a more complex case, Fig. 3c shows interleaved vision and text modalities forming a mixed structure. However, by spatially aggregating regions of the same modality, we observe that sparsity patterns can extend beyond local regions and often exhibit global extrapolation potential. The upper-left region in Fig. 3c exemplifies this, where the grid pattern, initially separated by textual boundaries, becomes consecutive after spatial clustering in both row and column dimensions. To validate this observation, we conducted a quantitative attention recall experiment on mixed-modality inputs, as detailed in §E.2.

3 MMInference

Following the analysis in §2, we propose MMInference to accelerate the pre-filling stage of long-context VLMs as shown in Fig. 4. The framework consists of three modules, covering both inter- and intra-modality sparse patterns: 1) the novel Grid sparse attention, together with the A-shape and Vertical-Slash patterns Jiang et al. (2024) forms the intra-modality attention; 2) Q-Boundary and 2D-Boundary mix-modality patterns; 3) Modality-aware sparse attention search algorithm. We first perform offline pattern search to identify different patterns for each attention head. Then we use online dynamic sparse approximation to build the sparse index, and finally we perform dynamic sparse computation using optimized GPU kernels.

Algorithm 1 Grid Head

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, stride space $s_g \in \phi_g$

 # Approximate stride and phase (last_q = 64)

$\hat{\boldsymbol{A}} \leftarrow \text{softmax}\left(\boldsymbol{Q}_{[-\text{last_q}:]}\boldsymbol{K}^\top/\sqrt{d} + \boldsymbol{m}_{\text{casual}}\right)$

 # Online search grid stride and phase

$\boldsymbol{b}_r, \leftarrow 0$

for $i \leftarrow 1$ to $|\phi_g|$ **do**

  if $\max(\text{view}(\hat{\boldsymbol{A}}, s_{g,i})) > \boldsymbol{b}_r$ **then**

    $s_g \leftarrow s_{g,i}, p_g \leftarrow \text{argmax}(\text{view}(\hat{\boldsymbol{A}}, s_{g,i}))$

    $\boldsymbol{b}_r \leftarrow \max(\text{view}(\hat{\boldsymbol{A}}, s_{g,i}))$

  end if

end for

 # Permute Q, K, V tensors

$\overline{\boldsymbol{Q}}, \overline{\boldsymbol{K}}, \overline{\boldsymbol{V}} \leftarrow \text{permute}(\boldsymbol{Q}), \text{permute}(\boldsymbol{K}), \text{permute}(\boldsymbol{V})$

 # Dynamic block sparse attention w/ FlashAttention (only the last and rightmost block)

$\boldsymbol{A} \leftarrow \text{softmax}\left(\text{sparse}(\overline{\boldsymbol{Q}}\overline{\boldsymbol{K}}^\top, s_g, p_g)/\sqrt{d}\right)$

 # Sparse mixed scores and values

$\boldsymbol{y} \leftarrow \text{sparse}(\boldsymbol{A}\overline{\boldsymbol{V}}, s_g, p_g)$

return $\boldsymbol{y}$

3.1 Grid Head in Multi-Modality

To better leverage the inductive bias in visual modalities (e.g., images, videos) and the vertical and horizontal structures in attention patterns, we propose a permutation-based dynamic sparse attention for grid head, as shown in Algo. 1.

Specifically, we first perform an online search to determine the stride and phase of grid pattern. Since only a view operation is applied to the approximate attention matrix $\hat{\boldsymbol{A}}$, the actual latency overhead remains minimal. Next, we use the identified grid stride and phase to permute the $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ tensors to compute sparse attention efficiently (see Fig. 3d). In our implementation, instead of explicitly permuting $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, we optimize computational efficiency by dynamically loading and writing these tensors within the kernel, minimizing the overhead associated with tensor transpositions. In addition to Grid sparse attention, we also employ A-shape and Vertical-Slash attention for intra-modality operation, see Appendix C.3 for more details.

3.2 Hybrid Modality Sparse Attention

As analyzed in §2 and illustrated in Fig. 3, modality boundaries exist in multi-modal LLMs. We classify these boundaries into four patterns: No-Boundary, K-Boundary, Q-Boundary, and 2D-Boundary. As the sparse index is continuous along the query dimension for both the No-Boundary and K-Boundary heads, we can directly apply the three intra-modality attention globally. However, for Q-Boundary and 2D-Boundary, MMInference uses a permutation-based approach to efficiently handle these modality boundaries.

Q-Boundary Head As shown in Fig.3b, Fig.3e, and §2.4, the Q-Boundary pattern shows a clear separation across modality, but the sparse distribution remains continuous within each modality. Building on this insight, we propose a row-wise permutation (Algorithm 2) that groups tokens of the same modality by permuting $\boldsymbol{Q}$, and then applies offline-optimized sparse attention (A-shape, Vertical-Slash, and Grid Head) for intra-modality processing. Note that we leverage the final segment of *each modality's* queries to dynamically approximate the sparse indices and extrapolate to the entire modality. This method enables flexibility in handling fragmented multi-modality inputs. Additionally, instead of explicitly permuting tensors, our implementation performs dynamic loading and writing inside the kernel for optimized efficiency.

Algorithm 2 Q-Boundary Head

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, modality type index $\boldsymbol{i}_m$, modality type set $m \in \phi_m$

 # Permute Q tensors based on modality
$\overline{\boldsymbol{Q}} \leftarrow \text{permute}\left(\boldsymbol{Q}, \boldsymbol{i}_m\right)$

 # Looping over the modalities in query dimension
$\boldsymbol{y} \leftarrow \boldsymbol{0}$
for $i \leftarrow 1$ to $|\phi_m|$ **do**

 # Intra-modality sparse attention for each modality w/ FlashAttention
$\boldsymbol{A}_{mi} \leftarrow \text{softmax}\left(\text{sparse}(\overline{\boldsymbol{Q}}_{mi}\boldsymbol{K}^\top, \boldsymbol{i}_{mi})/\sqrt{d}\right)$

$\boldsymbol{y}_{mi} \leftarrow \text{sparse}(\boldsymbol{A}_{mi}\boldsymbol{V})$

 # Update the modality output to the final output
$\boldsymbol{y} \leftarrow \boldsymbol{y}_{mi} \cup \boldsymbol{y}$
end for
return $\boldsymbol{y}$

Algorithm 3 2D-Boundary Head

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, modality type index $\boldsymbol{i}_m$, modality type set $m \in \phi_m$

 # Permute Q, K, V tensors based on modality
$\overline{\boldsymbol{Q}} \leftarrow \text{permute}\left(\boldsymbol{Q}, \boldsymbol{i}_m\right), \overline{\boldsymbol{K}} \leftarrow \text{permute}\left(\boldsymbol{K}, \boldsymbol{i}_m\right)$
$\overline{\boldsymbol{V}} \leftarrow \text{permute}\left(\boldsymbol{V}, \boldsymbol{i}_m\right)$

 # Looping over the modalities in pairs
$\boldsymbol{y} \leftarrow \boldsymbol{0}$
for $i \leftarrow 1$ to $|\phi_m|$ **do**
for $j \leftarrow 1$ to $|\phi_m|$ **do**

 # Dynamic sparse attention for each modality pair w/ FlashAttention
$\boldsymbol{m}_{mi,mj} \leftarrow \text{buildmask}(\boldsymbol{i}_{mi}, \boldsymbol{i}_{mj})$
$\boldsymbol{A}_{mi,mj} \leftarrow \text{softmax}($
$\text{sparse}(\overline{\boldsymbol{Q}}_{mi}\overline{\boldsymbol{K}}_{mj}^\top, \boldsymbol{i}_{mi}, \boldsymbol{i}_{mj})/\sqrt{d} + \boldsymbol{m}_{mi,mj})$

$\boldsymbol{y}_{mi,mj} \leftarrow \text{sparse}(\boldsymbol{A}_{mi,mj}\overline{\boldsymbol{V}}_{mj})$

 # Update the modality output to the final output

$\boldsymbol{y} \leftarrow \boldsymbol{y}_{mi,mj} \cup \boldsymbol{y}$
end for
end for
return $\boldsymbol{y}$

2D-Boundary Head Beyond Query-Boundary, there are attention heads that exhibit modality boundaries in both query and key dimensions, as shown in Fig. 3c. Given a query token, attention to key tokens from different modalities varies significantly, and queries from different modalities focus on keys in highly diverse patterns. To address 2D modality boundaries, we design a 2D permutation approach that groups $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ according to their modalities. This allows us to leverage intra-modality continuity to handle each part of 2D boundary pattern separately and efficiently. We further illustrate this approach in Fig. 3f and it detailed in Algorithm 3. Specifically, we perform permutation on both row- and column-wise for $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, and then iteratively traverse each modality pair to compute dynamic

sparse attention. The 2D-Boundary requires constructing an attention mask and searching for sparse patterns in cross-modality regions. For example, in Fig. 3f, we build modality boundary indices for Vision-to-Text (bottom-left) and Text-to-Vision (upper-right) attention. This mask index construction is implemented in Triton Tillet et al. (2019).

3.3 Modality-Aware Sparse Attention Search Algorithm

Due to modality boundaries in VLMs, we propose a modality-aware sparse attention pattern search algorithm (see Algorithm 4). The process unfolds in three steps: 1) intra-modality search within each modality following Jiang et al. (2024), 2) cross-modality search across all modality pairs, and 3) inter-modality search informed by the results of the first two steps.

Table 1: Performance (%) of different models and different methods on video understanding tasks evaluated at frames from 110 to 256.

Model	FLOPs	VideoDC	ActNet-QA	EgoSchema	Next-QA	PerceptionTest	VideoMME		Avg.
		test	test	test	mc	val	w/o sub.	w/ sub.	
Llava-Video-7B # Frames: 110; Total # tokens: 20,240									
Full Attention	100%	3.66	59.6	57.0	81.2	66.1	64.7	71.0	57.6
SF-fixed	4.8%	3.26	57.3	53.3	79.8	62.9	59.9	67.1	54.8
SF-strided	41.4%	3.45	58.5	56.1	80.6	64.4	61.4	68.5	56.1
A-shape	48.2%	3.56	56.0	51.6	79.8	65.7	54.4	65.6	53.8
Tri-shape	49.0%	3.58	59.3	54.5	80.3	66.1	63.6	70.1	56.7
VisionZip	35.2%	1.35	42.1	40.5	69.5	41.4	44.9	62.1	43.1
MInference	78.8%	3.64	59.6	57.0	80.6	66.1	64.6	71.0	57.5
Ours	47.3%	3.58	**59.8**	**57.1**	80.1	**66.2**	64.5	**71.8**	**57.6**
LongVILA-7B # Frames: 256; Total # tokens: 65,800									
Full Attention	100%	2.76	59.5	61.9	80.7	58.1	60.1	65.1	55.5
SF-fixed	2.2%	1.99	51.3	59.6	76.5	55.5	57.1	63.0	52.1
SF-strided	26.6%	2.58	56.0	61.4	76.7	55.5	53.6	59.2	52.2
A-shape	29.1%	2.75	56.6	60.9	75.0	55.3	49.1	59.6	51.3
Tri-shape	29.3%	2.63	58.1	62.0	77.8	56.2	59.3	63.3	54.2
VisionZip				OOM					
MInference	47.0%	2.77	59.7	62.2	79.1	57.8	60.0	65.2	55.2
Ours	31.8%	**2.84**	**60.2**	62.2	**79.4**	57.8	60.0	**65.5**	**55.4**
Qwen2.5-VL-7B-Instruct # Frames: 256; Total # tokens: 33,950									
Full Attention	100%	3.71	58.3	64.3	85.4	68.7	64.7	71.3	59.5
Ours	41.3%	**3.75**	58.0	63.9	84.9	**68.9**	**65.1**	70.9	59.4

4 EXPERIMENTS

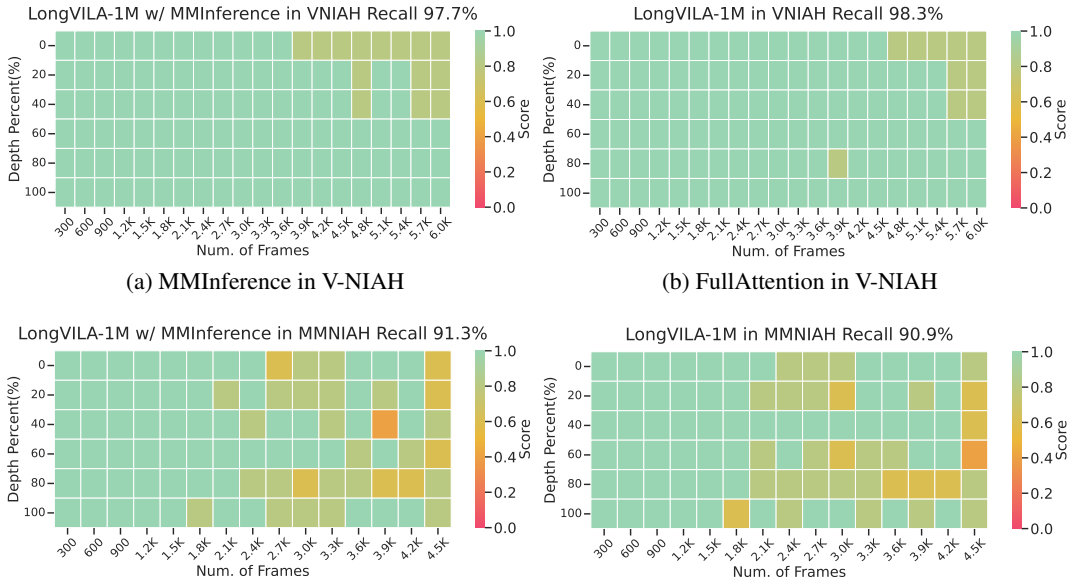
In this section, we address two key questions: (i) **How effective MMInference is?** We evaluate our method on three general long-video tasks: long-video understanding, Video Needle in a Haystack, and Video-Text Needle in a Haystack. These benchmarks cover long-video captioning, open-ended QA, multiple-choice QA, mixed-modality tasks, and retrieval tasks, providing a comprehensive assessment of MMInference's effectiveness across diverse long-video scenarios. (ii) **How efficient MMInference is?** We analyze end-to-end latency and its breakdown to thoroughly evaluate the efficiency of MMInference.

4.1 DATASET AND BASELINES

Implementation Details Our experiments are conducted on two state-of-the-art long-video VLMs: Llava-Video Zhang et al. (2024b) and LongVILA Chen et al. (2025). We follow the MInference experimental setup, configuring the corresponding search space while adopting optimal configurations from prior work for other methods. We adjust the local window sizes of A-shape and tri-shape patterns to align FLOPs with our method. For MInference, we adopt its optimal configuration, which results with FLOPs approximately twice as high as our method's in VLMs. Our implementation leverages Triton Tillet et al. (2019), FlashAttention Dao (2024), and dynamic sparse compiler PIT Zheng et al. (2023). For the Vertical-Slash and Grid Head patterns, we set $last_q = 64$. Latency experiments are performed on a single NVIDIA A100 using bfloat16, with greedy decoding to ensure stable results. Additional implementation details are provided in Appendix C.

Dataset Our evaluation uses the official metrics and scripts provided by these tasks. Additionally, we introduce a Mixed-Modality Needle in a Haystack (MM-NIAH) task to assess VLMs' retrieval capabilities on mixed-modality inputs. Dataset details are provided in Appendix D.

(i) Video Understanding Tasks: These include ActNet-QA Yu et al. (2019), EgoSchema Mangalam et al. (2023), Next-QA Xiao et al. (2021), PerceptionTest Patraucean et al. (2024), VideoDC Lab (2024), and VideoMME Fu et al. (2024). These benchmarks span five categories, covering tasks such as captioning and video question answering. Input lengths range from 110 frames (e.g., 20k) to 256 frames (e.g., 66k) in Llava-Video Zhang et al. (2024b) and LongVILA Chen et al. (2025).



(a) MMInference in V-NIAH (b) FullAttention in V-NIAH

(c) MMInference in MM-NIAH (d) FullAttention in MM-NIAH

Figure 5: V-NIAH Zhang et al. (2024a) and MM-NIAH results using LongVila-Qwen2-7B-1M Chen et al. (2025).

(ii) Video Needle in a Haystack (V-NIAH) Zhang et al. (2024a): A long-video retrieval task testing VLMs' performance with tokens of up to 6k frames (e.g., 1.1M tokens), where inserted images are placed at various positions.

(iii) Mixed-Modality Needle in a Haystack (MM-NIAH): To evaluate VLMs in mixed-modality scenarios, we construct a mix-modality version of NIAH. Specifically, 25% of the input consists of text segments inserted at the document level across different frames in long-video inputs, forming a mix-modality haystack. All other settings align with V-NIAH, including the multi-choice VQA task with randomly inserted images. This benchmark tests input lengths of up to 4.5k frames (e.g., 1.1M tokens).

Baselines We include five training-free sparse attention approaches, one visual token compression method, and also incorporate FlashAttention-2 Dao (2024) as a baseline. 1) SparseTransformer (Fixed) Child et al. (2019): Retains attention within each segment and allows all tokens to attend to the segment's initial tokens. 2) SparseTransformer (Strided) Child et al. (2019): Employs local windows with dilated attention. 3) A-Shape Xiao et al. (2024): Preserves only the sink token with local attention. 4) Tri-Shape LI et al. (2025); Acharya et al. (2024): Extends A-Shape by enabling full attention for all tokens to the last window's queries. 5) Vertical-Slash Pattern Jiang et al. (2024): Focuses on specific tokens (vertical lines) and tokens at fixed intervals (slash lines). 6) VisionZip Yang et al. (2024): A visual token compression method that reduces the number of visual tokens per frame by evaluating tokens based on their attention scores and discarding less important ones. Full details on implementation, hyperparameters, and illustrations for our baselines can be found in Appendix C.

4.2 Long Video Understanding

Table 1 presents the performance of different methods on video understanding tasks. The results show that: 1) Our method and MInference closely approximate full attention across all tasks while requiring only half the FLOPs of MInference. 2) Static sparse patterns, such as A-shape and Tri-shape, perform reasonably well on most tasks but experience a notable performance drop in multi-choice VQA tasks like EgoSchema. Additionally, the slight increase in query full attention in Tri-shape effectively improves performance. 3) Among SF patterns, the slash pattern better preserves

performance. Even when using SF-fixed with only 2%-5% of FLOPs, it still maintains strong performance on most tasks.

4.3 Video Needle In A Haystack

Fig. 5a, 5b, and 13 show the performance of different models on V-NIAH, revealing notable differences in handling long-context video retrieval as the number of processed frames increases: 1) Our method achieves results nearly identical to full attention. 2) A-shape struggles with mid-context information even at 300 frames, while Tri-shape maintains full performance until 3.9k frames (i.g. 700K tokens) before a sharp decline. 3) SF-fixed degrades at 2.1k frames (i.g. 350K tokens), while SF-strided surpasses Tri-shape, holding performance until 4.5k frames (i.g. 825K tokens). 4) MInference preserves VLM retrieval well, with only slight degradation beyond 4.8K frames.

4.4 Mixed-Modality Needle In A Haystack

Beyond V-NIAH, we introduce a mixed-modality NIAH test to evaluate the performance of different sparse methods on video-text inputs, in Fig. 5c, 5d, and 14. Mixed-modality inputs lead to more pronounced performance degradation across all methods. However, by incorporating inter-modality sparse patterns, our method maintains performance close to full attention, especially when compared to MInference and ours w/o inter-modality. Notably, Tri-shape and MInference show significant drops at 1.8k frames (i.g. 440K tokens) and 2.7k frames (i.g. 660K tokens).

5 Related Work

Long-Context Vision Language Models Recent VLMs have extended their context length to support long multi-modal inputs Zhang et al. (2024a); Chen et al. (2025); Wang et al. (2024c); Team et al. (2024), enabling applications such as long-video understanding Fu et al. (2024); Xiao et al. (2021); Wang et al. (2024a); Bai et al. (2025), multi-modal retrieval Zhang et al. (2024a), and multi-modal chain-of-thought reasoning Qwen (2024). For instance, Zhang et al. (2024a) transfer long-context capabilities from base LLMs to vision tasks, Chen et al. (2025) introduce multi-modal sequence parallelism to accelerate video fine-tuning, and Zhang et al. (2024b) emphasize the role of data calibration and synthetic data in boosting VLM performance.

Efficiency Optimization for VLMs While long-context VLMs achieve high accuracy, their high inference cost limits practical use in long-video scenarios. A common strategy is *vision token compression*—reducing video feature resolution by dropping or merging less important visual tokens Bolya et al. (2023); Chen et al. (2024); Shen et al. (2024); He et al. (2024); Tu et al. (2025); Weng et al. (2024); Wen et al. (2024). RNN-Transformer hybrids are also used Wang et al. (2024c) to balance efficiency and context length. However, these methods often assume inputs are long videos paired with short text, focusing solely on visual token optimization, while overlooking mixed-modality inputs critical for multi-turn interactions Huang et al. (2024). Recently, Xu et al. (2025b) applied dynamic sparse attention to long-context VLMs, but their approach ignores modality-specific inductive biases and is limited to single-modality video tasks.

6 Conclusion

We propose MMInference, a modality-aware permutation sparse attention method that accelerates long-context VLMs. It features permutation-based grid sparse attention, Q-boundary/2D-boundary patterns for mixed-modality boundaries, and a Modality-Aware Sparse Attention Search Algorithm. Our optimized GPU kernels enable end-to-end acceleration. Experiments on video understanding tasks, V-NIAH and MM-NIAH using Llava-Video and LongVila demonstrate that MMInference preserves full-attention performance while achieving up to 8.3× speedup at 1M tokens.

References

Shantanu Acharya, Fei Jia, and Boris Ginsburg. Star attention: Efficient llm inference over long sequences. *ArXiv preprint*, abs/2411.17116, 2024. URL https://arxiv.org/abs/2411.

17116.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *ArXiv preprint*, abs/2502.13923, 2025. URL https://arxiv.org/abs/2502.13923.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=c0chJTSbci.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *ICLR*, 2023.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *ArXiv preprint*, abs/2410.06158, 2024. URL https://arxiv.org/abs/2410.06158.

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. *ECCV*, pp. 19–35, 2024. doi: 10.1007/978-3-031-73004-7_2.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Yihui 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wCXAlfvCy6.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *ArXiv preprint*, abs/1904.10509, 2019. URL https://arxiv.org/abs/1904.10509.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *ArXiv preprint*, abs/2407.10759, 2024. URL https://arxiv.org/abs/2407.10759.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ICLR*, 2024.

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. *ArXiv preprint*, abs/2405.21075, 2024. URL https://arxiv.org/abs/2405.21075.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Tw9nfNyOMy.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6185–6194, 2023.

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 and kv cache compression. *ArXiv preprint*, abs/2410.08584, 2024. URL https://arxiv.org/abs/2410.08584.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *ArXiv preprint*, abs/2309.17080, 2023. URL https://arxiv.org/abs/2309.17080.

Minbin Huang, Yanxin Long, Xinchi Deng, Ruihang Chu, Jiangfeng Xiong, Xiaodan Liang, Hong Cheng, Qinglin Lu, and Wei Liu. Dialoggen: Multi-modal interactive dialogue system for multi-turn text-to-image generation. *ArXiv preprint*, abs/2403.08857, 2024. URL https://arxiv.org/abs/2403.08857.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *ArXiv preprint*, abs/2310.06825, 2023. URL https://arxiv.org/abs/2310.06825.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=fPBACAbqSN.

LMMs Lab. Video detail caption, 2024. URL https://huggingface.co/datasets/lmms-lab/VideoDetailCaption.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OfjIlbelrT.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *ArXiv preprint*, abs/2501.00574, 2025a. URL https://arxiv.org/abs/2501.00574.

Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, et al. Baichuan-omni-1.5 technical report. *ArXiv preprint*, abs/2501.15368, 2025b. URL https://arxiv.org/abs/2501.15368.

YUCHENG LI, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. SCBench: A KV cache-centric analysis of long-context methods. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gkUyYcY1W9.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=poE54GOq2l.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, et al. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *ArXiv preprint*, abs/2409.10516, 2024a. URL https://arxiv.org/abs/2409.10516.

Lei Liu, Xiaoyan Yang, Junchi Lei, Xiaoyang Liu, Yue Shen, Zhiqiang Zhang, Peng Wei, Jinjie Gu, Zhixuan Chu, Zhan Qin, et al. A survey on medical large language models: Technology, application, trustworthiness, and future directions. *ArXiv preprint*, abs/2406.03712, 2024b. URL https://arxiv.org/abs/2406.03712.

Liu Liu, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yufei Ding, and Yuan Xie. Dynamic sparse attention for scalable transformer acceleration. *IEEE Trans. Computers*, pp. 3165–3178, 2022. doi: 10.1109/TC.2022.3208206.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, pp. 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ACL*, pp. 12585–12602, 2024. doi: 10.18653/V1/2024.ACL-LONG.679.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *NeurIPS*, 2023.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adrià Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. *NeurIPS*, 2023.

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

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *ArXiv preprint*, abs/2405.07503, 2024. URL https://arxiv.org/abs/2405.07503.

Team Qwen. Dao, tri and haziza, daniel and massa, francisco and sizov, grigory, 2023. URL https://crfm.stanford.edu/2023/10/12/flashdecoding.html.

Team Qwen. Qvq: To see the world with wisdom, 2024. URL https://qwenlm.github.io/blog/qvq-72b-preview/.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. Sparq attention: Bandwidth-efficient llm inference. *ICML*, 2024.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *ArXiv preprint*, abs/2410.17434, 2024. URL https://arxiv.org/abs/2410.17434.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv preprint*, abs/2403.05530, 2024. URL https://arxiv.org/abs/2403.05530.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HMrcv7Q4Ub.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *ArXiv preprint*, abs/2409.12191, 2024a. URL https://arxiv.org/abs/2409.12191.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pp. 55–72. Springer, 2024b.

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *ArXiv preprint*, abs/2409.02889, 2024c. URL https://arxiv.org/abs/2409.02889.

Yuxin Wen, Qingqing Cao, Qichen Fu, Sachin Mehta, and Mahyar Najibi. Efficient vision-language models by summarizing visual tokens into compact registers. *ArXiv preprint*, abs/2410.14072, 2024. URL https://arxiv.org/abs/2410.14072.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. *ECCV*, pp. 453–470, 2024. doi: 10.1007/978-3-031-73414-4_26.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EytBpUGB1Z.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ICLR*, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. *CVPR*, pp. 9777–9786, 2021. doi: 10.1109/CVPR46437.2021.00965.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *ArXiv preprint*, abs/2503.20215, 2025a. URL https://arxiv.org/abs/2503.20215.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. Xattention: Block sparse attention with antidiagonal scoring. *ArXiv preprint*, abs/2503.16428, 2025b. URL https://arxiv.org/abs/2503.16428.

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. *ArXiv preprint*, abs/2412.04467, 2024. URL https://arxiv.org/abs/2412.04467.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. *AAAI*, pp. 9127–9134, 2019. doi: 10.1609/AAAI.V33I01.33019127.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, YX Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025.

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 preprint*, abs/2406.16852, 2024a. URL https://arxiv.org/abs/2406.16852.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *ArXiv preprint*, abs/2410.02713, 2024b. URL https://arxiv.org/abs/2410.02713.

Ningxin Zheng, Huiqiang Jiang, Quanlu Zhang, Zhenhua Han, Lingxiao Ma, Yuqing Yang, Fan Yang, Chengruidong Zhang, Lili Qiu, Mao Yang, et al. Pit: Optimization of dynamic sparse deep learning models via permutation invariant transformation. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 331–347, 2023.

A MODALITY-AWARE SPARSE ATTENTION SEARCH ALGORITHM

In Algorithm 4, we detail the procedure for selecting the optimal sparse attention pattern for each attention head under a constrained FLOPs budget. The algorithm jointly determines the best pattern and its configuration (e.g., stride size in grid attention, number of vertical/slash lines in VS pattern) to maximize accuracy. We first construct a kernel-aware search space, where all candidate patterns have comparable real-world FLOPs based on GPU kernel measurements—rather than theoretical estimates—to ensure practical efficiency.

We then evaluate each candidate using a reference example and select the configuration that maximizes attention recall, using the actual attention output as the objective. This recall-based scoring incorporates the V matrix and builds on FlashAttention Dao (2024), enabling end-to-end pattern selection with minimal memory overhead and improved performance.

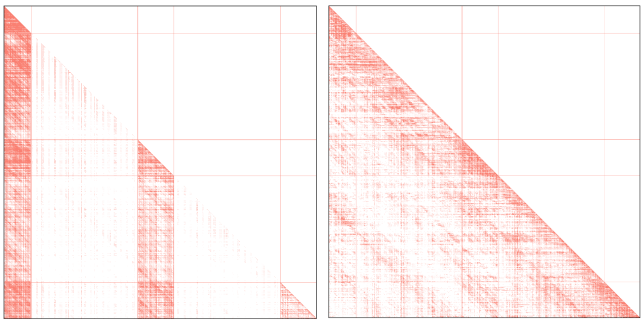
Algorithm 4 Modality-aware Sparse Attention Pattern Search

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, inter-modality search space ρ_{inter}, intra-modality search space ρ_{intra}, modality type set $m \in \phi_m$, optimized sparse pattern P

 # Intra-modality sparse attention pattern search
for $i \leftarrow 1$ to $|\phi_m|$ **do**
  $p_{mi} \leftarrow \text{KernelAwareSearch}\,(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, m_i)$
  $\text{P} \leftarrow \text{P} \cup p_{mi}$
end for

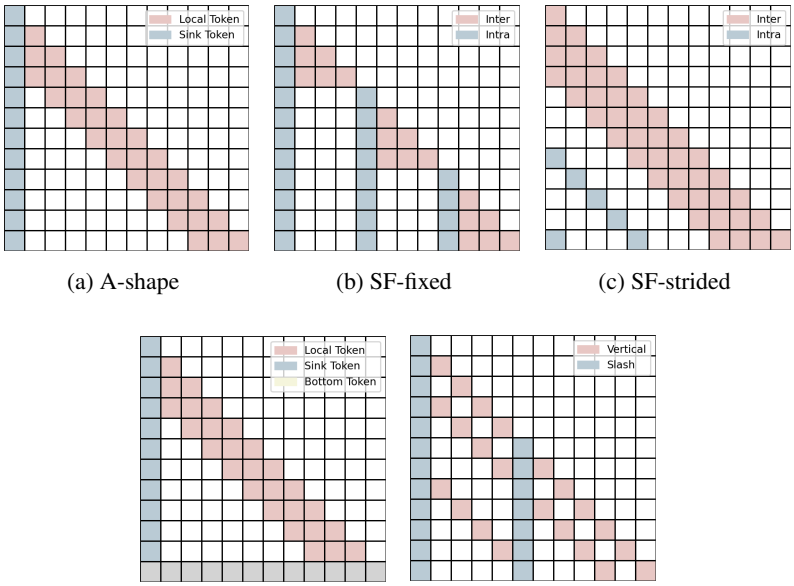
 # Cross-modality sparse attention pattern search
for $i \leftarrow 1$ to $|\phi_m|$ **do**
  for $j \leftarrow 1$ to $|\phi_m|$ **do**
    $p_{mi,mj} \leftarrow \text{KernelAwareSearch}\,(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, m_i, mj)$
    $\text{P} \leftarrow \text{P} \cup p_{mi,mj}$
  end for
end for

 # Inter-modality sparse attention pattern search
for $i \leftarrow 1$ to $|\rho_{\text{inter}}|$ **do**
  $p_i \leftarrow \text{argmin}\,(|\text{sparse}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, i) - \text{attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})|$
  $\text{P} \leftarrow \text{P} \cup p_i$
end for
return P



(a) K-Boundary pattern. (b) No-Boundary pattern.

Figure 6: Additional inter-modality sparse pattern.



(a) A-shape (b) SF-fixed (c) SF-strided

(d) Tri-shape (e) Vertical-Slash (MInference)

Figure 7: The baselines of sparse attention in our experiments.

B Pattern Analysis

B.1 Additional Mix-modality Pattern

In §2, we explain how the grid pattern naturally arises from the geometric structure of vision inputs. Fig. 6 further illustrates two additional patterns in the mixed-modality search space: the K-Boundary and No-Boundary patterns. Notably, both patterns incur no additional cost compared to pure intra-modality attention, as their sparse indices can be computed across all rows without extra computation.

B.2 Additional Sparse Attention Pattern Visualization

We further analyze the sparse patterns in Qwen2.5-VL Wang et al. (2024a) with dynamic resolution inputs and in VideoChat-Flash Li et al. (2025a) under visual token compression, across both video benchmark and mixed-modality inputs, as shown in Fig.17 and Fig.18.

C Experiment Details

C.1 Vision Language Models

We use two state-of-the-art VLMs in our experiments: LongVILA Chen et al. (2025) and Llava-Video Zhang et al. (2024b). Llava-Video supports varying numbers of frames (32, 64, 110) for video understanding, and as reported, performance improves with more frames. Thus, we adopt the 110-frame variant for benchmarking. For LongVILA, we use the 256-frame version (LongVILA-256Frame) with a 128K context length for video understanding benchmarks, and the 1M-token version (LongVILA-1M), designed for retrieval tasks, for the V-NIAH evaluation.

C.2 Baselines

We include five sparse attention baselines in our experiments: A-shape Xiao et al. (2024), SF-fixed Child et al. (2019), SF-strided Child et al. (2019), Tri-shape LI et al. (2025), MInference Jiang

Table 2: Hyperparameters detail of baselines.

Method	Hyperparameters
A-shape	Sink = 128, Local = 4096
SF-fixed	Local = token_per_frame, vline_stride = token_per_frame
SF-strided	Local = token_per_frame, vline_stride = token_per_frame
Tri-shape	Sink = 128, Local = 4096, Bottom = 128
MInference	Vertical_size $\in \{1000, 2000, 4000\}$, Slash_size $\in \{1024, 2048, 4096, 6144\}$
VisionZip	dominant = 54, contextual = 10

et al. (2024), and VisionZip Yang et al. (2024). Fig. 7 illustrates the attention patterns of these baselines.

While VisionZip Yang et al. (2024) is primarily a visual token compression method—compressing vision tokens using attention scores from the vision encoder before passing them to the LLM—it is included for comparison as it reduces FLOPs in the pre-filling stage and offers insight into token compression approaches.

C.3 A-SHAPE AND VERTICAL-SLASH

A-shape and Vertical-Slash are used for intra-modality attention, alongside our newly proposed Grid pattern.

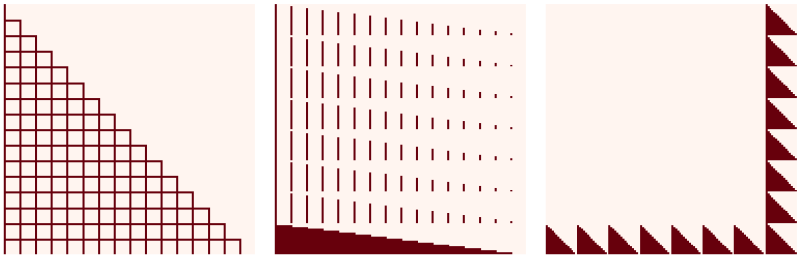
At inference time, we estimate the attention matrix online to dynamically determine the spatial layout of sparse indices, conditioned on the assigned pattern and actual input. Sparse attention is then computed using our optimized GPU kernels. Note that while the masks for Vertical-Slash and Grid patterns are dynamically generated, A-shape uses a static mask, incurring no additional overhead beyond sparse computation.

A-shape head. A-shape is a static sparse pattern that includes the first seven initial tokens along with a local attention window.

Vertical-Slash head. Due to the continuity of vertical and slash lines, we matmul the last query vector $\boldsymbol{Q}_{[-\text{last_q}:]}$ and key vector $\boldsymbol{K}$ to produce the estimated attention matrix $\hat{\boldsymbol{A}}$, which, in turn, is used to determine the indices for the vertical $\boldsymbol{i}_v$ and slash $\boldsymbol{i}_s$ lines. After obtaining the sparse indices for the vertical and slash lines, we convert them into a sparse format $\boldsymbol{i}_{vs}$. Using these sparse indices, we perform block-sparse calculations of the attention weights and attention output.

C.4 PERMUTATION FOR THE GRID PATTERN AND ACROSS MODALITY

We illustrate how the permutation is applied to the Grid pattern and the Q-boundary and 2D-boundary patterns in Fig. 8 and Fig. 8.



(a) Before Permutation (b) Row-wise Permutation (c) Column-wise Permutation

Figure 8: Permutation for the Grid Pattern. (a) Before permutation. (b) Row-wise permutation. (c) Column-wise permutation.

(a) Mix-modality (b) Q-wise Permutation (c) K-wise Permutation

Figure 9: Permutation for mix-modality context. (a) Mix-modality. (b) Q-wise permutation. (c) K-wise permutation.

Attention Type	**Parameters**
Grid Attention	(frame_stride, True, False, False, 1024) (frame_stride, False, True, False, 1024) (frame_stride, False, False, True, 1024) (frame_stride, True, True, False, 1024) (frame_stride, False, True, True, 1024) (frame_stride, True, True, True, 1024) (stride, True, False, False, 1024) (stride, False, True, False, 1024) (stride, False, False, True, 1024) (stride, True, True, False, 1024) (stride, False, True, True, 1024) (stride, True, True, True, 1024)
A-shape	(128, 1024) (128, 2048) (128, 4096)
Vertical-Slash	(1000, 1024) (1000, 2048) (2000, 2048) (1000, 3096) (2000, 3096) (1000, 4096) (2000, 4096) (3500, 200) (1000, 2500)

Table 3: The search space for each attention pattern: 1) Grid Attention: (stride, use hline, use vline, use slash, max stride); 2) A-shape: (sink, local); 3) Vertical-Slash: (vertical size, slash size)

C.5 Search Space

Following Jiang et al. (2024), we set the target FLOPs t to be the same as 1k global tokens and 4k local window tokens in the *A-shape* pattern. Additionally, we use only one sample as our calibration set from the egoschema task with no more than 25K tokens, which exhibits strong generalization and stability across different lengths and domains. The search time is approximately 15 minutes on a single A100. This pattern search is individually conducted for each model: Llava-Video-7B, LongVila-256Frame, and LongVila-1M. The search space is shown in Table 3.

D BENCHMARK DETAILS

We evaluate our method on several video understanding benchmarks that test different aspects of video comprehension:

EgoSchema EgoSchema Mangalam et al. (2023) is a diagnostic benchmark for very long-form video language understanding, structured as a multiple-choice question answering task. The benchmark requires models to answer questions about egocentric videos by selecting from given options (labeled A through E). The evaluation can be performed either on the full set via submission to an evaluation server, or on a released subset of 500 questions for direct scoring.

Video-MME Video-MME Fu et al. (2024) is a comprehensive multi-modal evaluation benchmark that tests MLLMs across diverse video types and temporal dimensions. It spans 6 primary visual domains with 30 subfields and includes videos ranging from 11 seconds to 1 hour in duration. The benchmark comprises 900 videos totaling 254 hours, with 2,700 manually annotated question-answer pairs. It evaluates models' ability to process not just video frames but also integrated multi-modal inputs like subtitles and audio.

NExT-QA NExT-QA Xiao et al. (2021) focuses on advancing video understanding from basic description to explaining temporal actions. It features both multiple-choice and open-ended QA tasks that target three key aspects: causal action reasoning, temporal action reasoning, and common scene comprehension. The benchmark is specifically designed to evaluate models' ability to reason about actions beyond superficial scene descriptions.

Perception Test The Perception Test Patraucean et al. (2023) perce evaluates perception and reasoning skills across video, audio, and text modalities. It contains 11.6k real-world videos with an average length of 23 seconds, featuring perceptually interesting situations. The benchmark tests four key skills (Memory, Abstraction, Physics, Semantics) and various types of reasoning (descriptive, explanatory, predictive, counterfactual). Videos are densely annotated with six types of labels: multiple-choice QA, grounded video QA, object tracks, point tracks, temporal action segments, and sound segments.

ActivityNet-QA ActivityNet-QA Yu et al. (2019) is a large-scale VideoQA dataset consisting of 58,000 QA pairs on 5,800 complex web videos derived from the ActivityNet dataset. The benchmark is fully annotated and designed to test models' understanding of complex web videos through question answering. Unlike automatically generated datasets, ActivityNet-QA features human-annotated questions and answers, making it particularly valuable for evaluating real-world video understanding capabilities.

Video Detail Description (VideoDC) VideoDC Lab (2024) focuses on comprehensive video understanding through detailed descriptions. The benchmark consists of question-answer pairs generated with GPT-3.5, where questions prompt for detailed descriptions focusing on main subjects, their actions, and background scenes. The evaluation assesses the quality and completeness of video descriptions generated by models.

E ADDITIONAL EXPERIMENTS RESULTS

E.1 LATENCY

Fig. 10 and 16 present end-to-end and kernel-level latency across different context sizes. The grid pattern significantly outperforms the vertical-slash pattern in sparsity, achieving a 2–3× speedup even at 1M tokens. Additionally, the grid pattern achieves an end-to-end speedup of 8.3× and a kernel-level speedup of 12×.

E.2 ANALYSIS

Transition of Sparse Patterns Across Modalities Since LLMs and VLMs exhibit different sparse patterns, we examine the interplay between the Grid and Vertical-Slash pattern. As shown in Fig. 11,

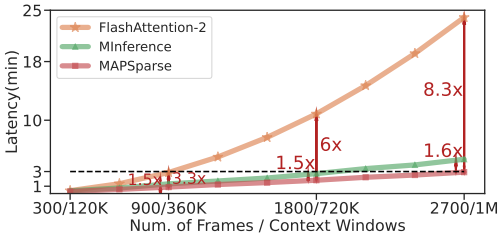
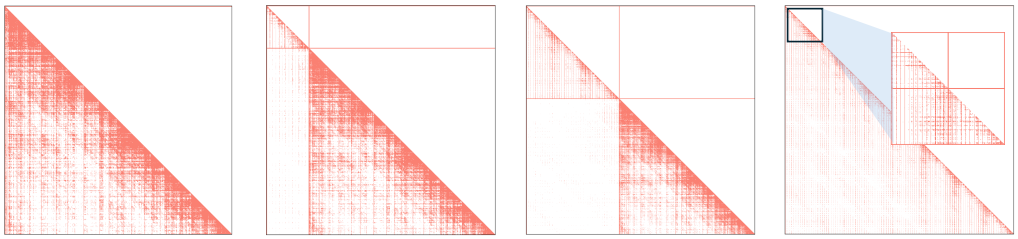


Figure 10: End-to-End Latency.



(a) All Textual Context (b) Visual Context Inserted (c) More Visual Context (d) All Visual Context

Figure 11: Transition of sparse patterns from textual context to visual context. (a) The vertical-slash pattern for all textual context. (b) Grid pattern appears when visual modality is appended. (c) Grid pattern dominates.

Llava-Video-7B primarily uses Vertical-Slash pattern for purely textual inputs. However, once a visual input is appended, it transitions to a Grid pattern to capture the geometric structure of the visual content. This shift occurs at the modality boundary, creating a more structured arrangement of vertical and horizontal intervals. Such behavior highlights the need for distinct sparsity strategies in visual and mixed-modality contexts, rather than simply reusing sparse patterns from LLMs for VLMs.

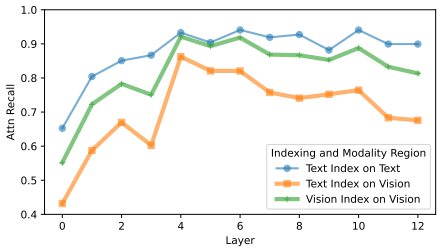


Figure 12: The sparse index does not effectively extrapolate from text to the visual modality. However, an index built within the same modality can generalize across modality boundaries.

Sparse Index Across Modalities In Fig. 12, the sparse index achieves high recall for textual regions but fails to generalize to visual ones. To address this, we construct a sparse index from the visual modality and evaluate it on separate visual segments, each separated by modality boundaries. Remarkably, this approach extrapolates effectively across all visual segments, even when interspersed with textual boundaries. As shown in Fig. 12, the sparse index achieves high recall in the textual but fails to generalize to the visual. To address this, we construct a sparse index using the visual modality and evaluate it across distinct regions of the visual modality, separated by modality boundaries. Remarkably, this approach successfully extrapolates to all visual regions even when interrupted by text-induced boundaries.

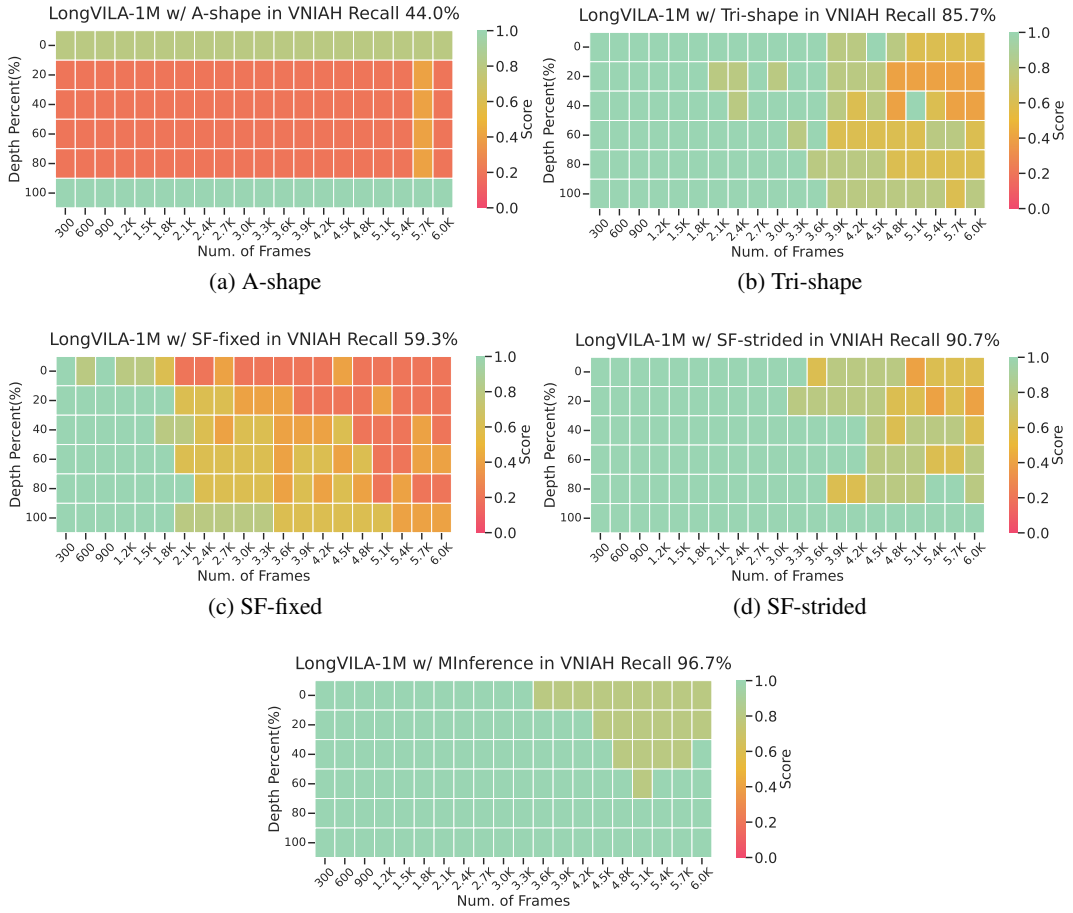
Integrate with token compression methods As shown in Table 4, our method integrates seamlessly with token compression techniques, enabling near-lossless performance while supporting longer or higher-resolution video inputs. Specifically, VideoChat-Flash reduces tokens per frame from 196 to 16 at the ViT stage, while our method further applies sparse attention in the LLM decoder. Results demonstrate strong performance retention across benchmarks.

Table 4: Performance (%) on video understanding tasks based on VideoChat-Flash Li et al. (2025a) at frames 512 with 8k tokens.

Model	VideoDC	ActNet-QA	EgoSchema	Next-QA	PerceptionTest	VideoMME		Avg.
	test	test	test	mc	val	w/o sub.	w/ sub.	
VideoChat-Flash	3.21	53.6	57.0	81.2	69.1	63.2	70.5	56.8
w/ MMInference	3.19	54.3	57.3	79.8	69.1	63.0	70.2	56.7

E.3 Additional Video Needle In A Haystack Results

we further present the results of the Video Needle In A Haystack task with our baselines. The results of our method and full atttenton is shown in Fig. 5.

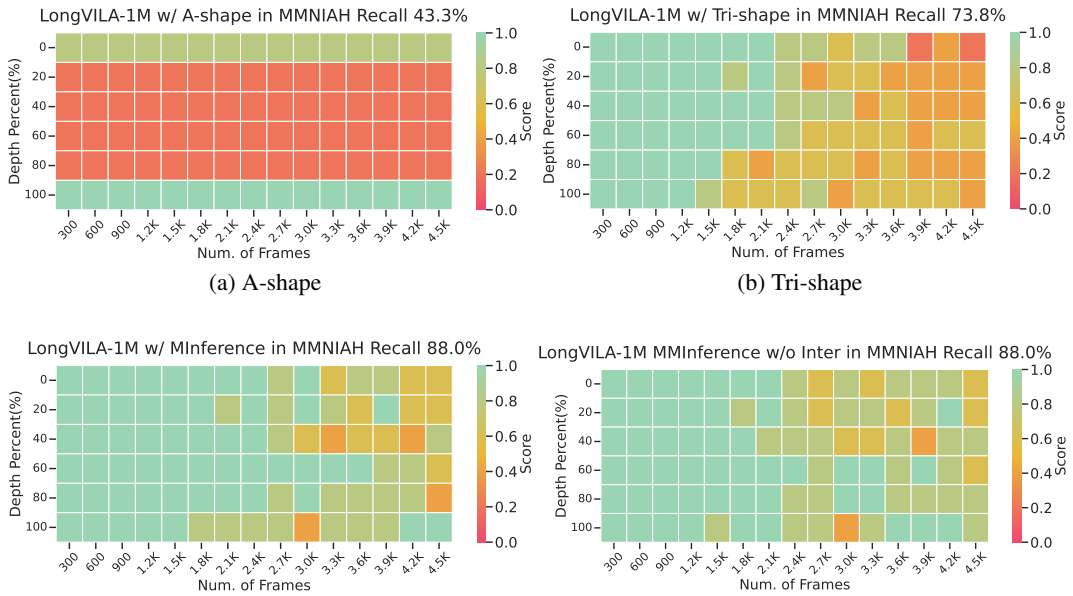


(a) A-shape (b) Tri-shape (c) SF-fixed (d) SF-strided (e) MInference

Figure 13: Video Needle In A Haystack Zhang et al. (2024a) results using LongVila-Qwen2-7B-1M Chen et al. (2025).

E.4 Additional Mixed-Modality Needle In A Haystack Results

We further present the results of the Mixed-Modality Needle In A Haystack task with our baselines and the inter-modality variant of our method. The results of full atttenton and MMInference are shown in Fig. 5.



(a) A-shape (b) Tri-shape

(c) MInference (d) MMInference w/o Inter-modality

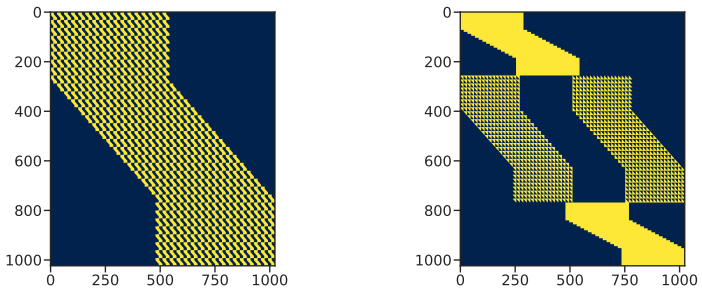
Figure 14: Mixed-Modality Needle In A Haystack results using LongVila-Qwen2-7B-1M Chen et al. (2025).

E.5 Latency Breakdown

As shown in Fig. 16, we present the micro-benchmark results of various sparse attention methods across different context lengths.

E.6 VS Pattern vs. Grid Pattern

Both VS pattern and Grid pattern achieve strong performance on video understanding and V-NIAH tasks. However, due to the grid attention pattern observed in VLMs, the overlap between blocks covered by diagonal lines in the VS pattern is minimal, reducing sparsity within the kernel. This explains why VS pattern exhibits significantly higher latency compared to Grid pattern. Additionally, leveraging permutation-based optimization effectively reduces the number of blocks involved in kernel computation, thereby lowering latency while maintaining comparable performance.



(a) Natten (b) Permutated Natten

Figure 15: Permutation-based implementation of 2D/3D sliding window attention Hassani et al. (2023) enables efficient sparse attention optimization for DiT architectures.

F SPARSE ATTENTION IN DIT

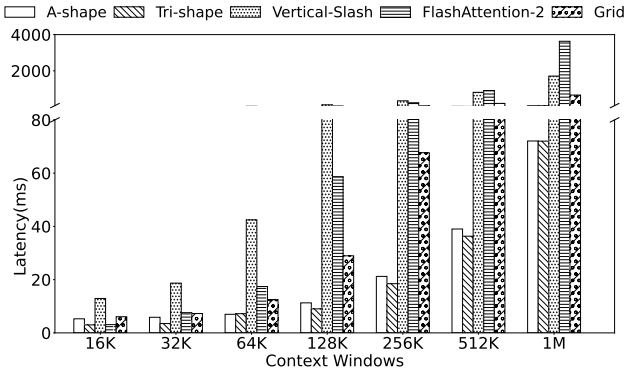


Figure 16: The latency breakdown of a single attention kernel for four sparse attention patterns and FlashAttention Dao (2024) across different context windows in a single A100, including the index time for dynamic sparse approximation and building dynamic sparsity. At 1M tokens, the latency for Grid is 358ms.

Recently, many efficient DiT methods Hassani et al. (2023); Xi et al. (2025); Zhang et al. (2025); Xu et al. (2025b) have adopted sparse attention to accelerate long video generation. We note that these methods can also benefit from permutation-based transformations to achieve kernel-efficient implementations. For example, the 2D/3D sliding window attention in NATTEN can be converted into dense tensor core computation via permutation, as illustrated in Fig. 15. Similarly, the temporal head in Sparse VideoGen Xi et al. (2025) and the anti-diagonal structure in xAttention Xu et al. (2025b) can be restructured through permutation to enable sparse loading with dense computation, significantly speeding up DiT inference, especially in long-context scenarios.

G KERNEL IMPLEMENTATION

As shown in Algorithms 5, 6, and 7, we provide implementation details of the FlashAttention-based kernels. The Grid-shape kernel in Algorithm 5 integrates block-sparse FlashDecoding Qwen (2023), which sparsifies the query loading, with block-sparse FlashAttention-2, which sparsifies the key loading. The Q-Boundary kernel in Algorithm 6 introduces sparsity along the query dimension using FlashAttention-2 Dao (2024), while the 2D-Boundary kernel in Algorithm 7 applies sparsity along both the query and key dimensions.

Algorithm 5 Grid-Shape Flash Attention

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, block size B, stride size σ, query start index s_q, key start index s_k

Scale $\tau \leftarrow \sqrt{\frac{1}{d_h}}$

Initialize $\boldsymbol{O} \leftarrow (0)^{S \times d_h} \in \mathbb{R}^{S \times d_h}$

 # Parallelized in GPU

 # Sparse load in $\boldsymbol{Q}$ using FlashDecoding

for $i \leftarrow 1$ to N_σ **do**

  Load $\boldsymbol{Q}_{\text{chip}} \leftarrow \boldsymbol{Q}^{[i \times B:(i+1) \times B] \times \sigma + s_q} \in \mathbb{R}^{B \times d_h}$

  Initialize $\boldsymbol{O}_{\text{chip}} \leftarrow (0)^{B \times d_h} \in \mathbb{R}^{B \times d_h}$

  Initialize $\boldsymbol{m} \leftarrow (-\inf)^B \in \mathbb{R}^B$

  Initialize $\boldsymbol{l} \leftarrow (0)^B \in \mathbb{R}^B$

  # Loop in $\boldsymbol{K}$

  for $j \leftarrow 1$ to M **do**

    Load $\boldsymbol{K}_{\text{chip}} \leftarrow \boldsymbol{K}^{[j \times B:(j+1) \times B]} \in \mathbb{R}^{B \times d_h}$

    Load $\boldsymbol{V}_{\text{chip}} \leftarrow \boldsymbol{V}^{[j \times B:(j+1) \times B]} \in \mathbb{R}^{B \times d_h}$

    $\boldsymbol{S} \leftarrow \tau \boldsymbol{Q}_{\text{chip}} \boldsymbol{K}_{\text{chip}}^T$

    $\boldsymbol{S} \leftarrow \text{mask}(\boldsymbol{S})$

    $\boldsymbol{m}_{new}^i \leftarrow \max(\boldsymbol{m}^i, \text{rowmax}(\boldsymbol{S})) \in \mathbb{R}^B$

    $\boldsymbol{S} \leftarrow \boldsymbol{S} - \boldsymbol{m}_{new}^i$

    $\boldsymbol{P} \leftarrow \exp(\boldsymbol{S})$

    $\boldsymbol{l}_{new}^i \leftarrow \text{rowsum}(\boldsymbol{S}))$

    $\boldsymbol{\alpha} \leftarrow \exp(\boldsymbol{m}^i - \boldsymbol{m}_{new}^i)$

    $\boldsymbol{l}^i \leftarrow \boldsymbol{\alpha} \boldsymbol{l}^i + \boldsymbol{l}_{new}^i$

    $\boldsymbol{O}_{\text{chip}} \leftarrow \boldsymbol{\alpha} \boldsymbol{O}_{\text{chip}} + \boldsymbol{P} \boldsymbol{V}_{\text{chip}}$

  end for

  # Write outputs

  $\boldsymbol{O}_{\text{chip}} \leftarrow \text{diag}(\boldsymbol{l}^i)^{-1} \boldsymbol{O}_{\text{chip}}$

  Save $\boldsymbol{O}_i \leftarrow \boldsymbol{O}_{\text{chip}}$

end for

 # Sparse load in $\boldsymbol{K}$ using FlashAttention

for $i \leftarrow 1$ to N **do**

  Load $\boldsymbol{Q}_{\text{chip}} \leftarrow \boldsymbol{Q}^{[i \times B:(i+1) \times B]} \in \mathbb{R}^{B \times d_h}$

  Initialize $\boldsymbol{O}_{\text{chip}} \leftarrow (0)^{B \times d_h} \in \mathbb{R}^{B \times d_h}$

  Initialize $\boldsymbol{m} \leftarrow (-\inf)^B \in \mathbb{R}^B$

  Initialize $\boldsymbol{l} \leftarrow (0)^B \in \mathbb{R}^B$

  # Loop in $\boldsymbol{K}$

  for $j \leftarrow 1$ to M_σ **do**

    Load $\boldsymbol{K}_{\text{chip}} \leftarrow \boldsymbol{K}^{[j \times B:(j+1) \times B] \times \sigma + \sigma \times s_k} \in \mathbb{R}^{B \times d_h}$

    Load $\boldsymbol{V}_{\text{chip}} \leftarrow \boldsymbol{V}^{[j \times B:(j+1) \times B] \times \sigma + \sigma \times s_k} \in \mathbb{R}^{B \times d_h}$

    $\boldsymbol{S} \leftarrow \tau \boldsymbol{Q}_{\text{chip}} \boldsymbol{K}_{\text{chip}}^T$

    $\boldsymbol{S} \leftarrow \text{mask}(\boldsymbol{S})$

    $\boldsymbol{m}_{new}^i \leftarrow \max(\boldsymbol{m}^i, \text{rowmax}(\boldsymbol{S})) \in \mathbb{R}^B$

    $\boldsymbol{S} \leftarrow \boldsymbol{S} - \boldsymbol{m}_{new}^i$

    $\boldsymbol{P} \leftarrow \exp(\boldsymbol{S})$

    $\boldsymbol{l}_{new}^i \leftarrow \text{rowsum}(\boldsymbol{S}))$

    $\boldsymbol{\alpha} \leftarrow \exp(\boldsymbol{m}^i - \boldsymbol{m}_{new}^i)$

    $\boldsymbol{l}^i \leftarrow \boldsymbol{\alpha} \boldsymbol{l}^i + \boldsymbol{l}_{new}^i$

    $\boldsymbol{O}_{\text{chip}} \leftarrow \boldsymbol{\alpha} \boldsymbol{O}_{\text{chip}} + \boldsymbol{P} \boldsymbol{V}_{\text{chip}}$

  end for

  # Write outputs

  $\boldsymbol{O}_{\text{chip}} \leftarrow \text{diag}(\boldsymbol{l}^i)^{-1} \boldsymbol{O}_{\text{chip}}$

  Save $\boldsymbol{O}_i \leftarrow \boldsymbol{O}_{\text{chip}}$

end for

Algorithm 6 Q-Boundary Flash Attention

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, block size B, modality index $\boldsymbol{I}_m$, sparse attention kernel $\mathbf{Op}_m$

Scale $\tau \leftarrow \sqrt{\frac{1}{d_h}}$

Initialize $\boldsymbol{O} \leftarrow (0)^{S \times d_h} \in \mathbb{R}^{S \times d_h}$

 # Loop modality and parallelized in GPU

for $m \in \{\text{text}, \text{vision}, ..., \}$ **do**

  for $i \leftarrow 1$ to N_m **do**

    Load index $\boldsymbol{I}_{\text{chip}} \leftarrow \boldsymbol{I}_m^{[i \times B:(i+1) \times B]} \in \mathbb{R}^B$

    Load $\boldsymbol{Q}_{\text{chip}} \leftarrow \boldsymbol{Q}^{\boldsymbol{I}_{\text{chip}}} \in \mathbb{R}^{B \times d_h}$

    Initialize $\boldsymbol{O}_{\text{chip}} \leftarrow (0)^{B \times d_h} \in \mathbb{R}^{B \times d_h}$

    Initialize $\boldsymbol{m} \leftarrow (-\inf)^B \in \mathbb{R}^B$

    Initialize $\boldsymbol{l} \leftarrow (0)^B \in \mathbb{R}^B$

    # Loop in $\boldsymbol{K}$ using modality sparse attention

    $\boldsymbol{O}_{\text{chip}}, \boldsymbol{m}, \boldsymbol{l} \leftarrow \mathbf{Op}_m(\boldsymbol{Q}_{\text{chip}}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{O}_{\text{chip}}, \boldsymbol{m}, \boldsymbol{l})$

    # Write outputs w/ modality index

    $\boldsymbol{O}_{\text{chip}} \leftarrow \text{diag}(\boldsymbol{l}^i)^{-1} \boldsymbol{O}_{\text{chip}}$

    Save $\boldsymbol{O}_i^{\boldsymbol{I}_{\text{chip}}} \leftarrow \boldsymbol{O}_{\text{chip}}$

  end for

end for

Algorithm 7 2D-Boundary Flash Attention

Input: $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{S \times d_h}$, block size B, modality index $\boldsymbol{I}_m$, sparse attention kernel $\mathbf{Op}_m$

Scale $\tau \leftarrow \sqrt{\frac{1}{d_h}}$

Initialize $\boldsymbol{O} \leftarrow (0)^{S \times d_h} \in \mathbb{R}^{S \times d_h}$

 # Loop modality and parallelized in GPU

for $m_q \in \{\text{text}, \text{vision}, ..., \}$ **do**

  for $i \leftarrow 1$ to $N_{m,q}$ **do**

    Load index $\boldsymbol{I}_{\text{chip},q} \leftarrow \boldsymbol{I}_{m,q}^{[i \times B:(i+1) \times B]} \in \mathbb{R}^B$

    Load $\boldsymbol{Q}_{\text{chip}} \leftarrow \boldsymbol{Q}^{\boldsymbol{I}_{\text{chip},q}} \in \mathbb{R}^{B \times d_h}$

    Initialize $\boldsymbol{O}_{\text{chip}} \leftarrow (0)^{B \times d_h} \in \mathbb{R}^{B \times d_h}$

    Initialize $\boldsymbol{m} \leftarrow (-\inf)^B \in \mathbb{R}^B$

    Initialize $\boldsymbol{l} \leftarrow (0)^B \in \mathbb{R}^B$

    # Loop in $\boldsymbol{K}$ and modality

    for $m_k \in \{\text{text}, \text{vision}, ..., \}$ **do**

      for $j \leftarrow 1$ to $M_{m,k}$ **do**

        Load index $\boldsymbol{I}_{\text{chip},k} \leftarrow \boldsymbol{I}_{m,k}^{[j \times B:(j+1) \times B]} \in \mathbb{R}^B$

        Load $\boldsymbol{K}_{\text{chip}} \leftarrow \boldsymbol{K}^{\boldsymbol{I}_{\text{chip},k}} \in \mathbb{R}^{B \times d_h}$

        Load $\boldsymbol{V}_{\text{chip}} \leftarrow \boldsymbol{V}^{\boldsymbol{I}_{\text{chip},k}} \in \mathbb{R}^{B \times d_h}$

        $\boldsymbol{O}_{\text{chip}}, \boldsymbol{m}, \boldsymbol{l} \leftarrow \mathbf{Op}_m(\boldsymbol{Q}_{\text{chip}}, \boldsymbol{K}_{\text{chip}}, \boldsymbol{V}_{\text{chip}}, \boldsymbol{O}_{\text{chip}}, \boldsymbol{m}, \boldsymbol{l})$

      end for

    end for

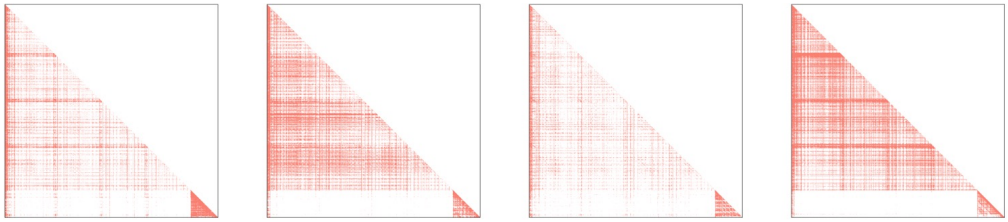
    # Write outputs w/ modality index

    $\boldsymbol{O}_{\text{chip}} \leftarrow \text{diag}(\boldsymbol{l}^i)^{-1} \boldsymbol{O}_{\text{chip}}$

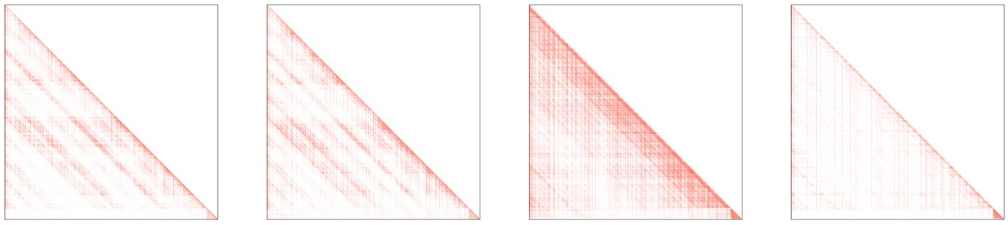
    Save $\boldsymbol{O}_i^{\boldsymbol{I}_{\text{chip},q}} \leftarrow \boldsymbol{O}_{\text{chip}}$

  end for

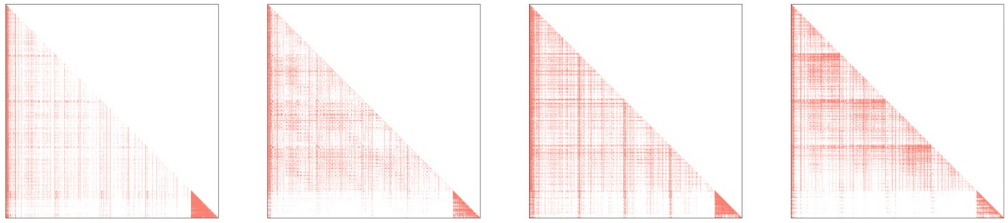
end for



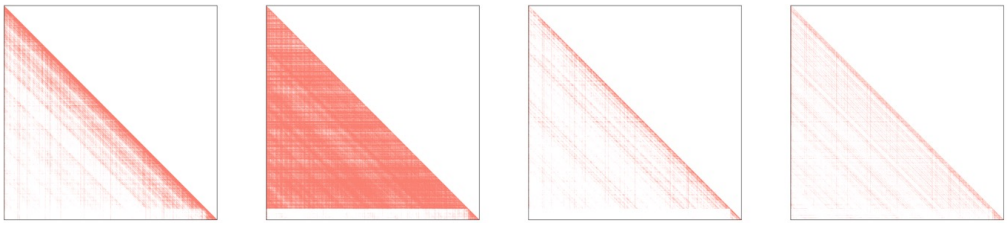
(a) Qwen2.5-VL on EgoSchema



(b) VideoChat on EgoSchema

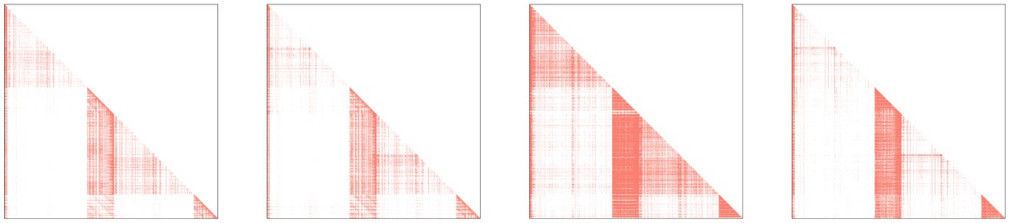


(c) Qwen2.5-VL on VideoMME

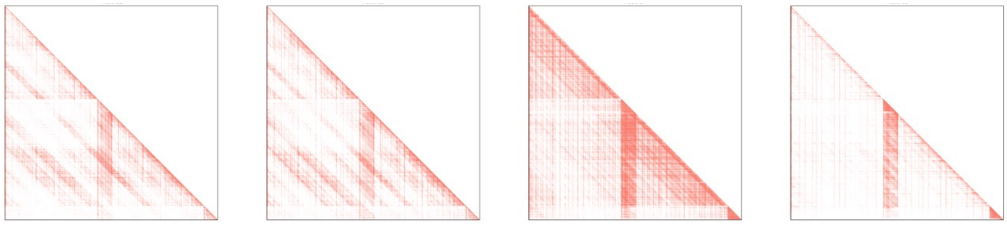


(d) VideoChat on VideoMME

Figure 17: Visualization of sparse attention patterns in Qwen2.5-VL with dynamic resolution input and VideoChat-Flash with visual token compression across different benchmarks.



(a) Qwen2.5-VL on Mix-modality



(b) VideoChat on Mix-modality

Figure 18: Visualization of sparse attention patterns in Qwen2.5-VL with dynamic resolution input and VideoChat-Flash with visual token compression with mix-modality inputs.