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

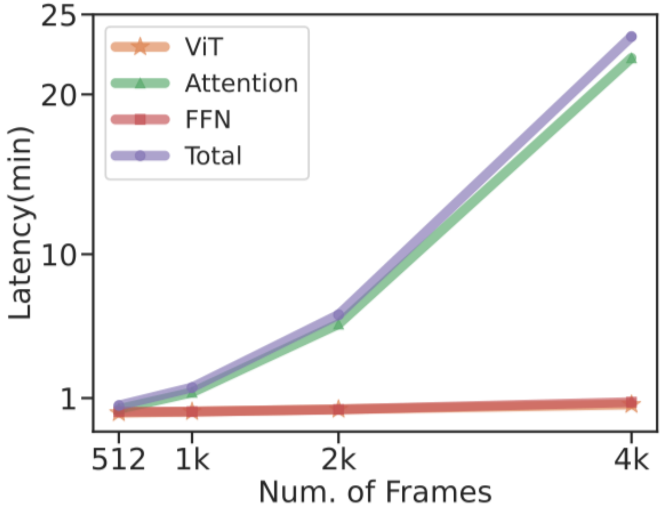
Microsoft Corporation, [◊]University of Surrey

https://aka.ms/MMInference

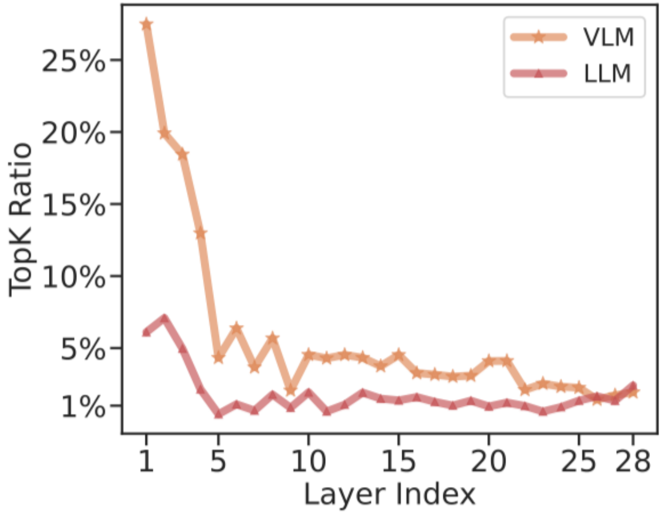
Microsoft Research

Observation 1: VLMs are also dynamically sparse.

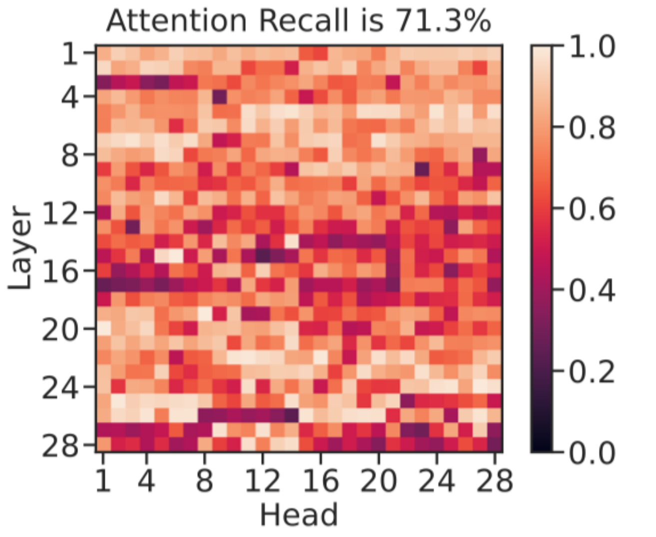
- ❑ Multi-modality Attention is **Dynamically Sparse**
- ❑ However, VLMs exhibit significantly **lower sparsity** than text-only LLMs (95% attention recall requires 5.78% vs. 1.78%). Still, 52.3% of heads need to recall less than 2% of attention.



(a) VLMs’ attention incurs heavy cost.



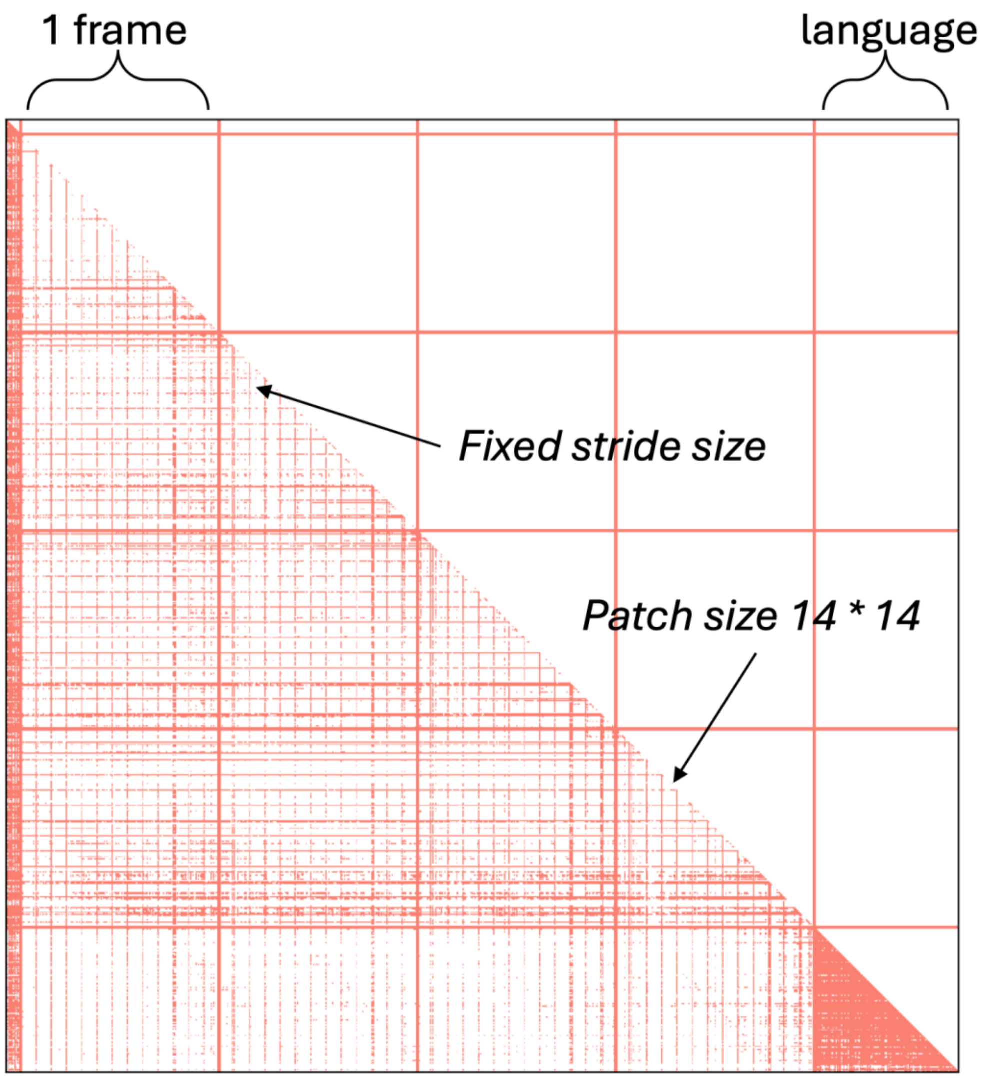
(b) VLMs’ attention is sparse.



(c) Sparsity of VLMs’ attention is dynamic.

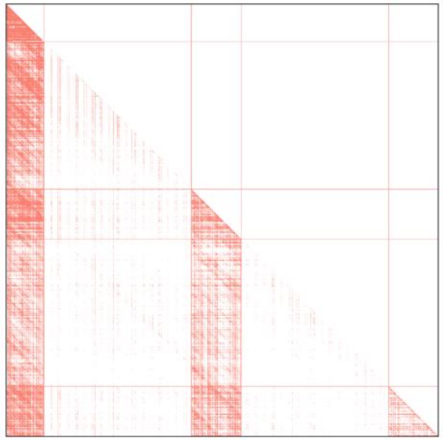
Observation 2: Grid Head in VLMs

- Local tokens in **temporal** and **spatial** dimensions are evenly distributed within the attention map.
- Stride and starting position vary with context, the horizontal and vertical lines are evenly spaced and often symmetrical.

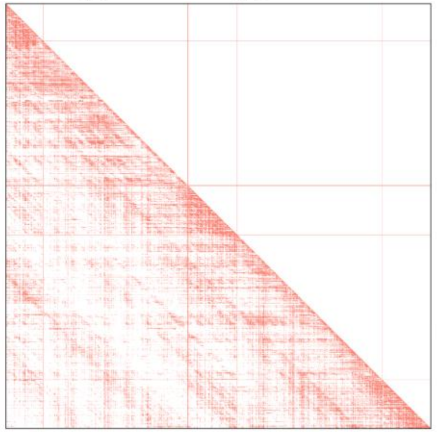


Observation 3: Modality Boundaries in Multi-Modal Input

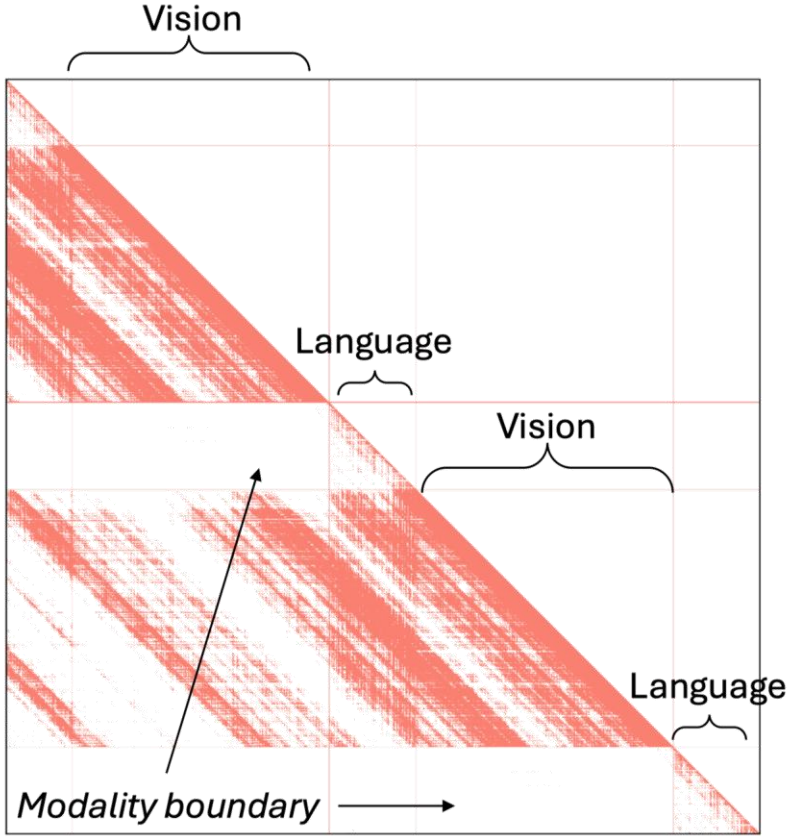
- ❑ 1) Intramodality consistency; 2) Modality-separated continuity



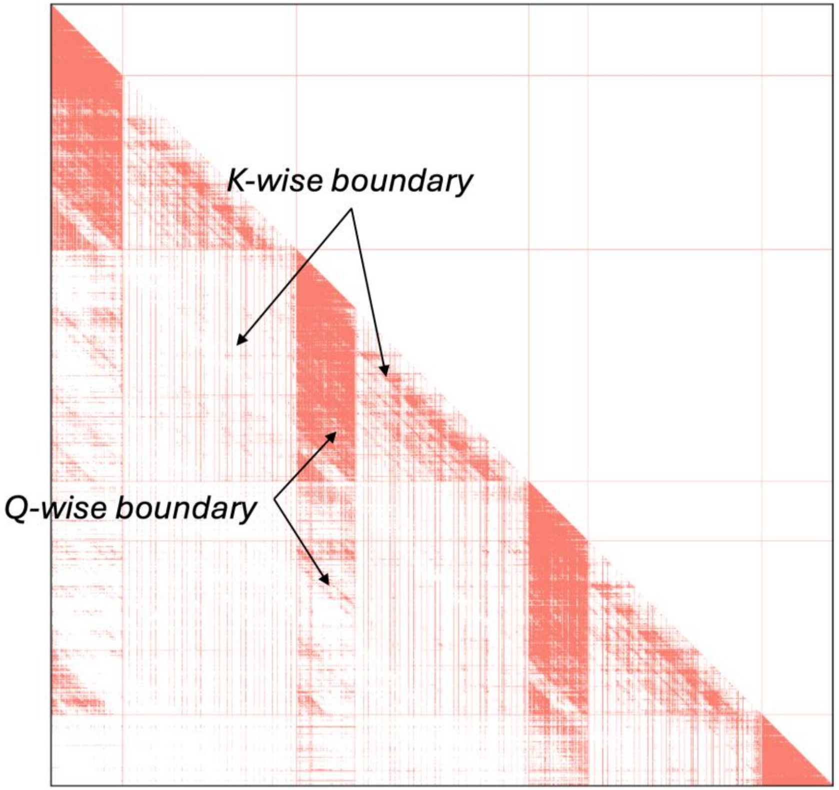
(a) K-Boundary pattern.



(b) No-Boundary pattern.



(b) Q-Boundary pattern.



(c) 2D-Boundary pattern.

Observation 4: Sparse Distributions Continuity Across Boundaries

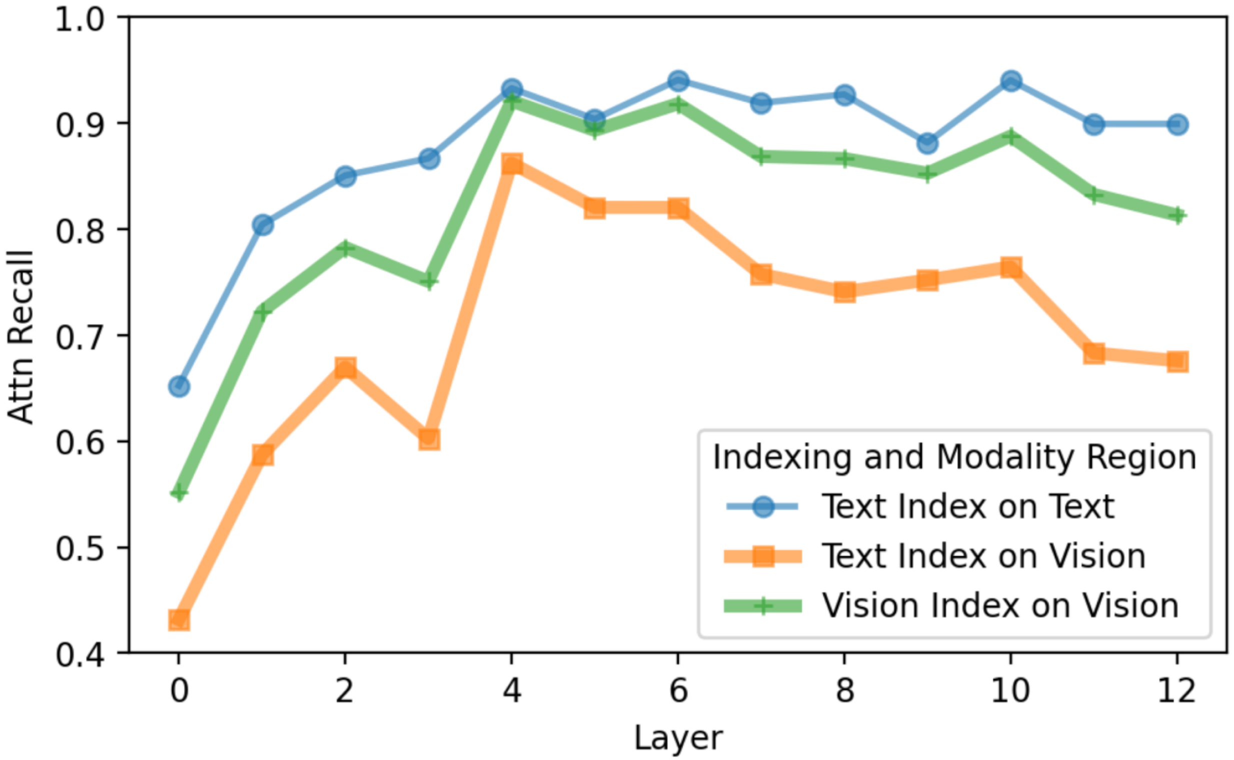
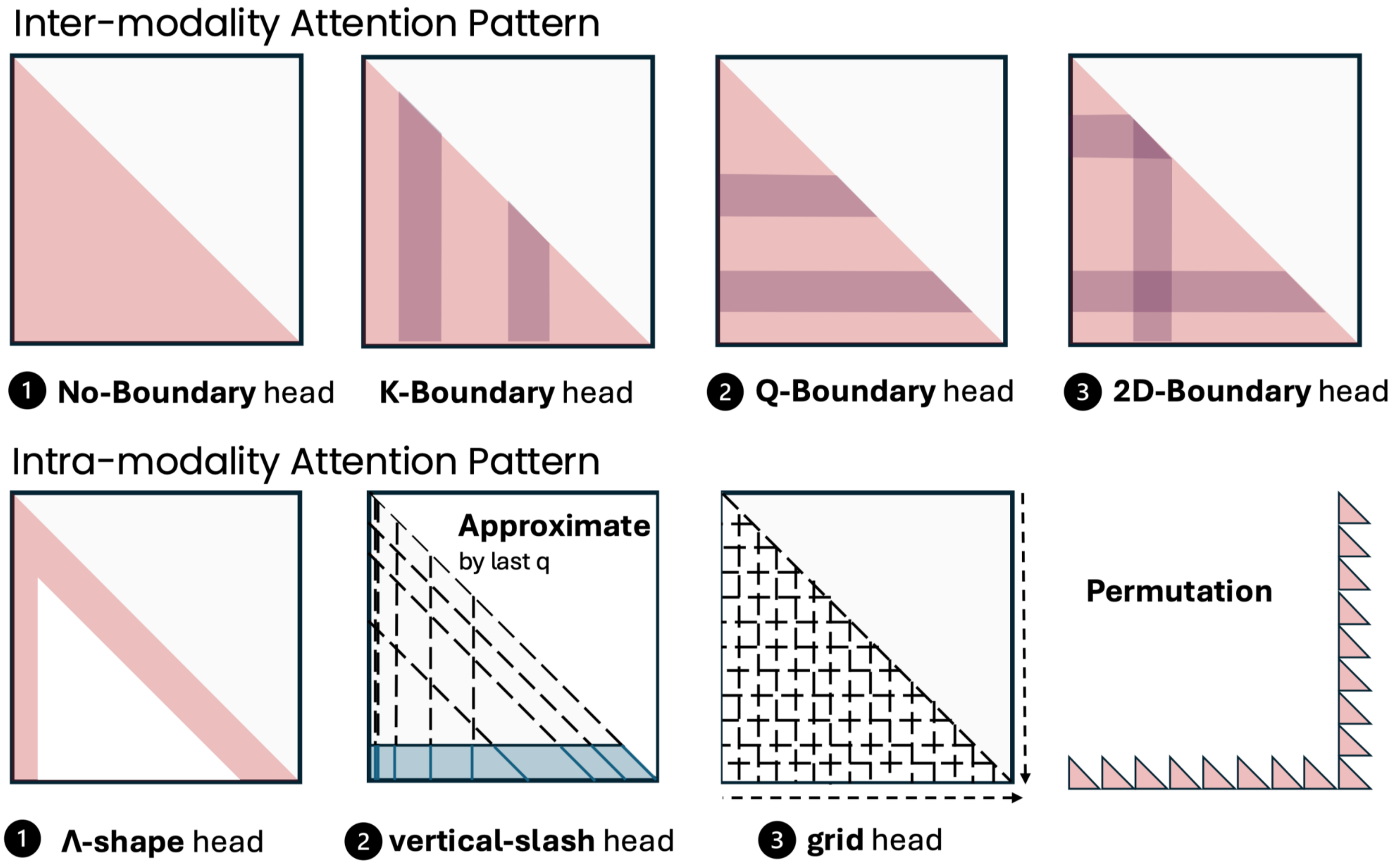
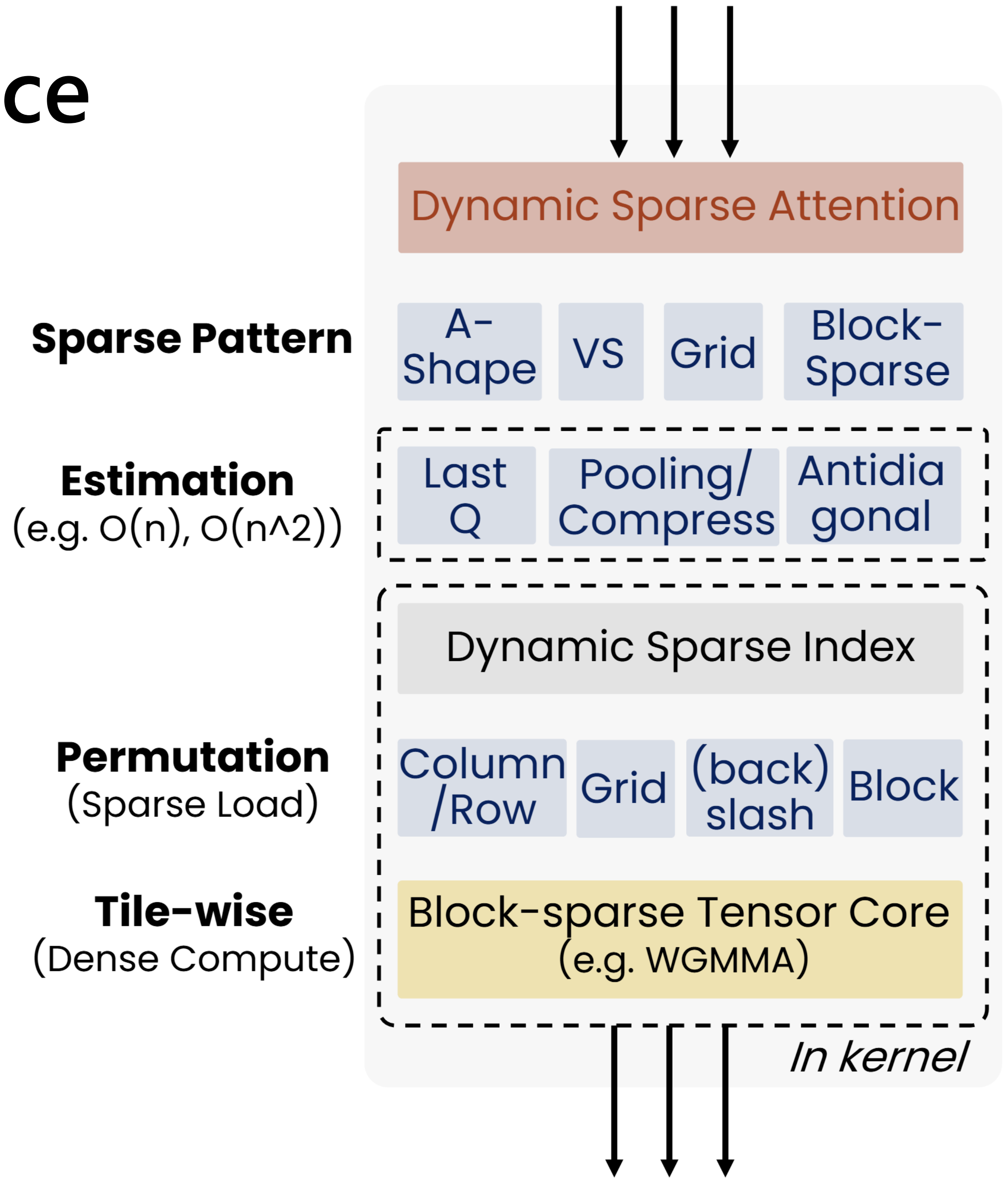


Figure 7: The sparse index does not effectively extrapolate from text to the visual modality. However, an index built within the same modality can generalize across modality boundaries.

MMInference



MMInference



MMInference: Grid Head in Multi-Modality

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)

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

 # Online search grid stride and phase

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

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

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

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

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

  end

end for

 # Permute Q, K, V tensors

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

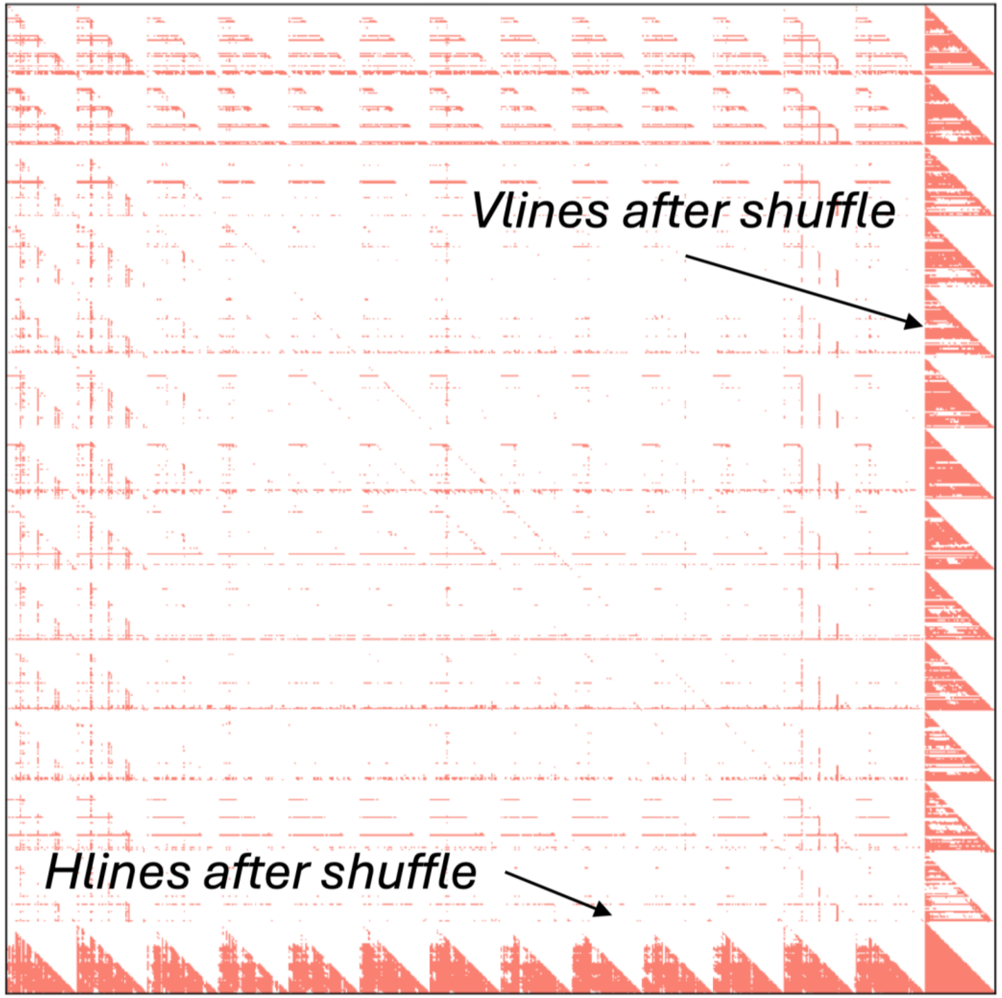
 # Final dynamic sparse attention scores w/ FlashAttention (only the last and rightmost block)

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

 # Sparse mixed scores and values

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

return $\boldsymbol{y}$

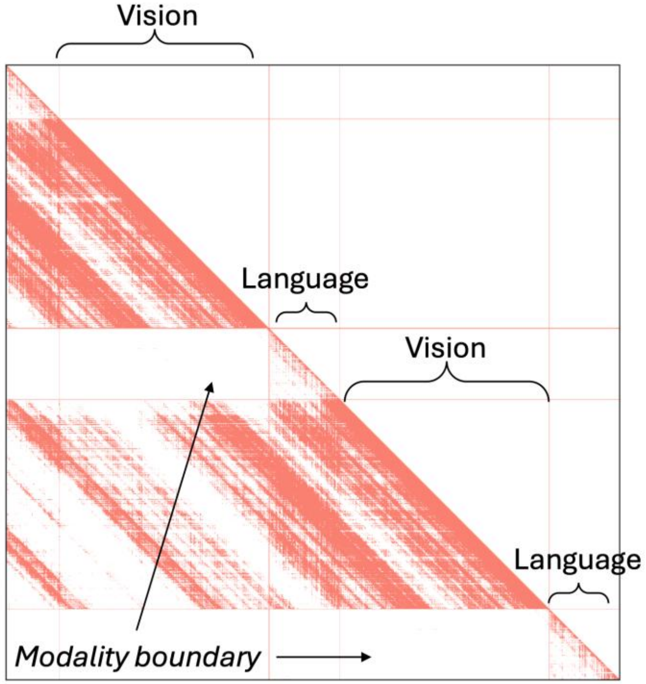


MMInference: Q-Boundary pattern

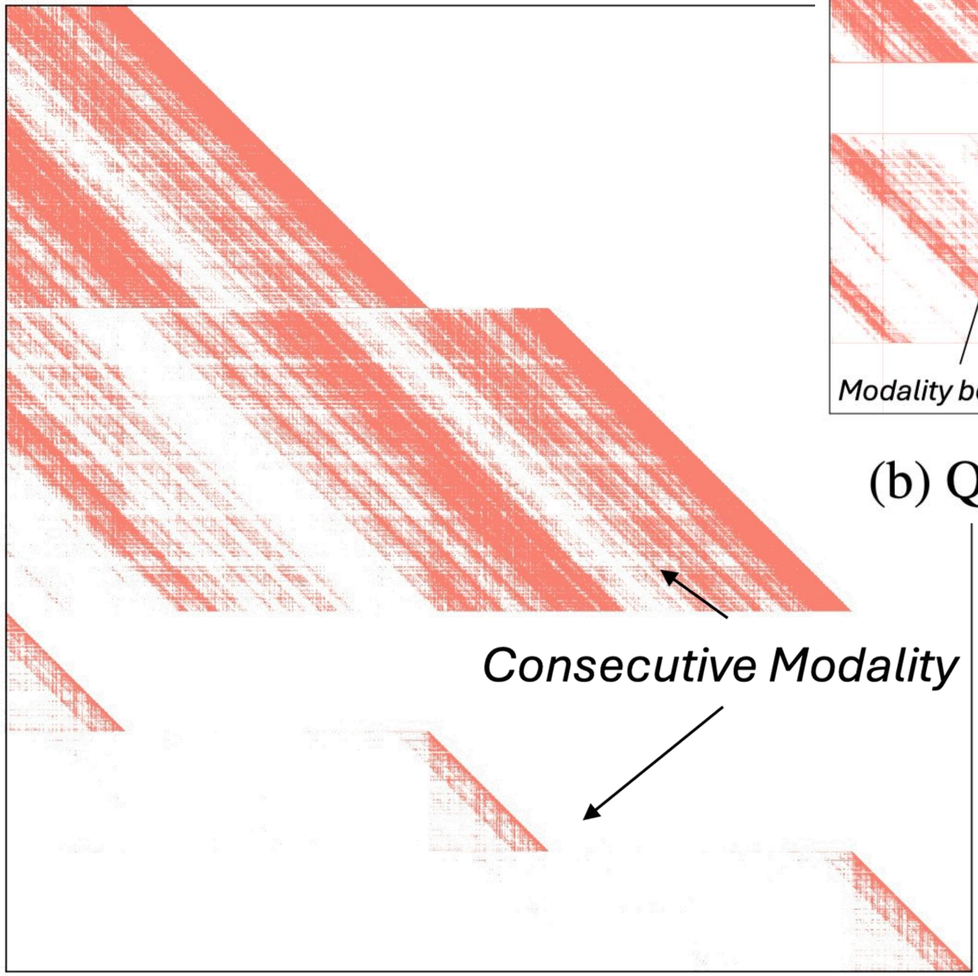
Algorithm 2 Q-Boundary Head

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
$\overline{\boldsymbol{Q}} \leftarrow \text{permute}\,(\boldsymbol{Q}, \boldsymbol{i}_m)$
 # Looping over the modalities in query dimension
$\boldsymbol{y} \leftarrow \boldsymbol{0}$
for $i \leftarrow 1$ to $|\phi_m|$ **do**
  # Intra-modality sparse attention computation 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 $\boldsymbol{y}$



(b) Q-Boundary pattern.



(e) Permuted Q-Boundary pattern.

MMInference: 2D-Boundary pattern

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 computation for each modality pair w/ FlashAttention

    $\boldsymbol{m}_{mi,mj} \leftarrow \text{buildmask}(\boldsymbol{i}_{mi}, \boldsymbol{i}_{mj})$

    $\boldsymbol{A}_{mi,mj} \leftarrow \text{softmax}($

    $\text{sparse}(\overline{\boldsymbol{Q}}_{mi}\overline{\boldsymbol{K}}_{mj}^{\top}, \boldsymbol{i}_{mi}, \boldsymbol{i}_{mj})/\sqrt{d} + \boldsymbol{m}_{mi,mj})$

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

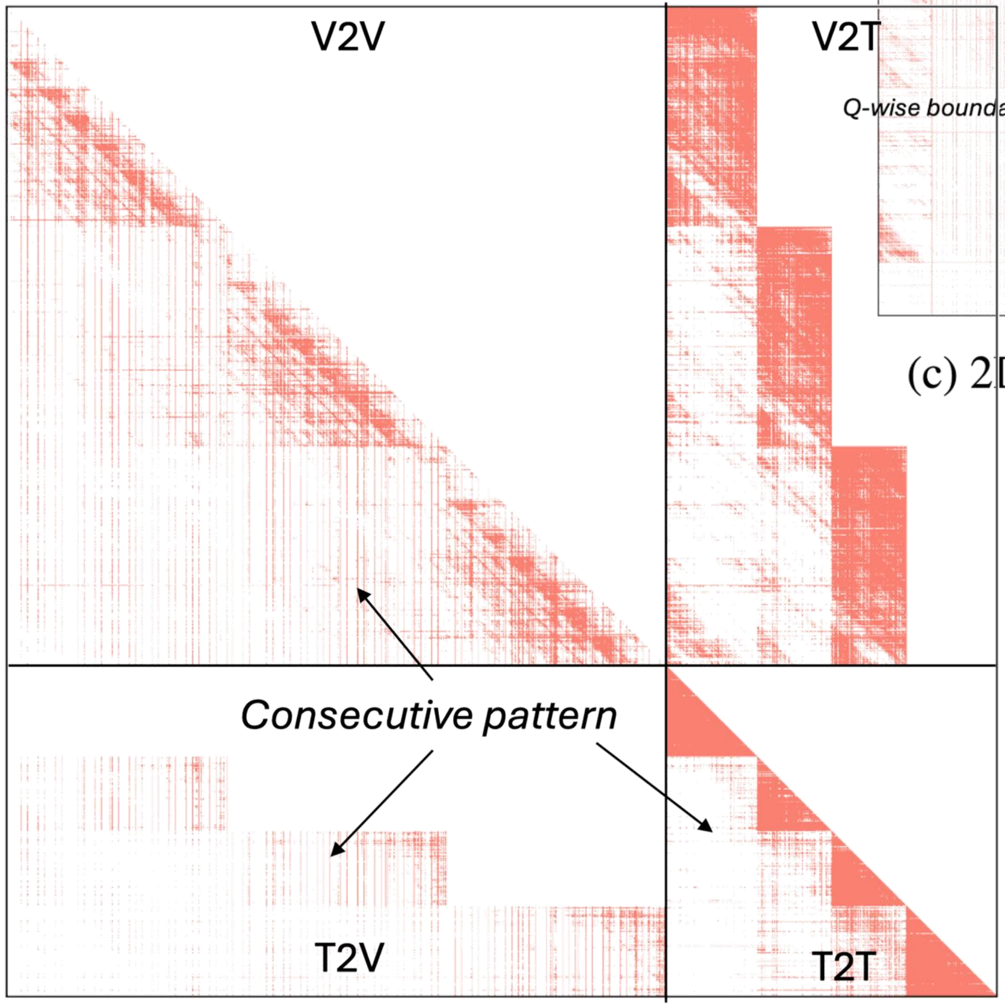
    # Update the modality output to the final output

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

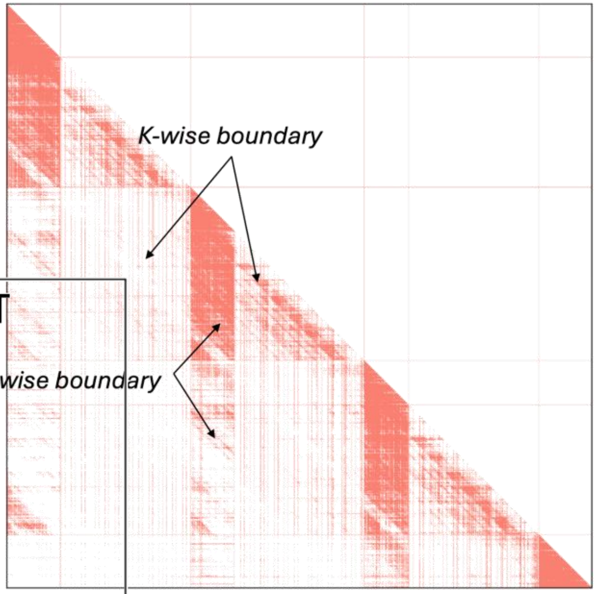
  end for

end for

return $\boldsymbol{y}$



(f) Permuted 2D-Boundary pattern.



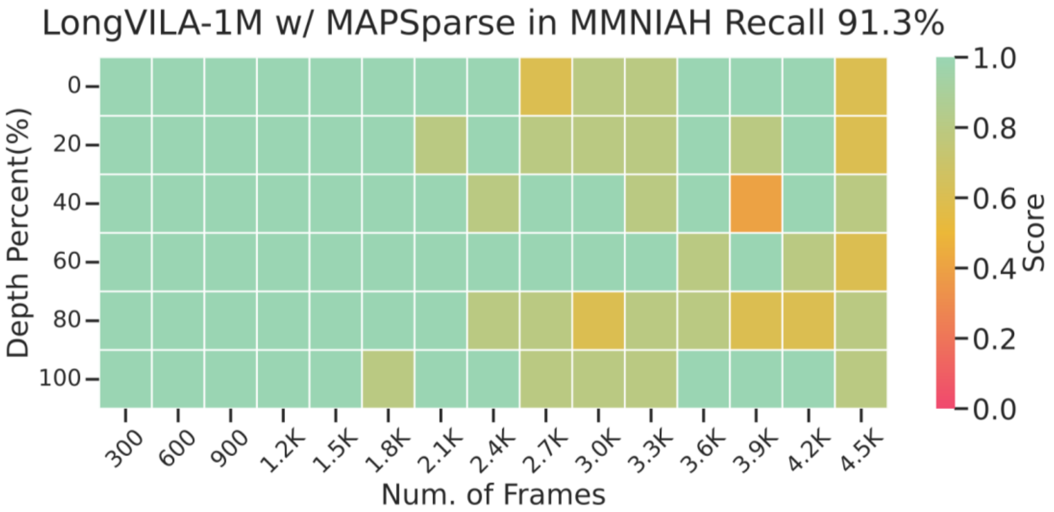
(c) 2D-Boundary pattern.

How effective is MMInference? Long-Video Benchmark

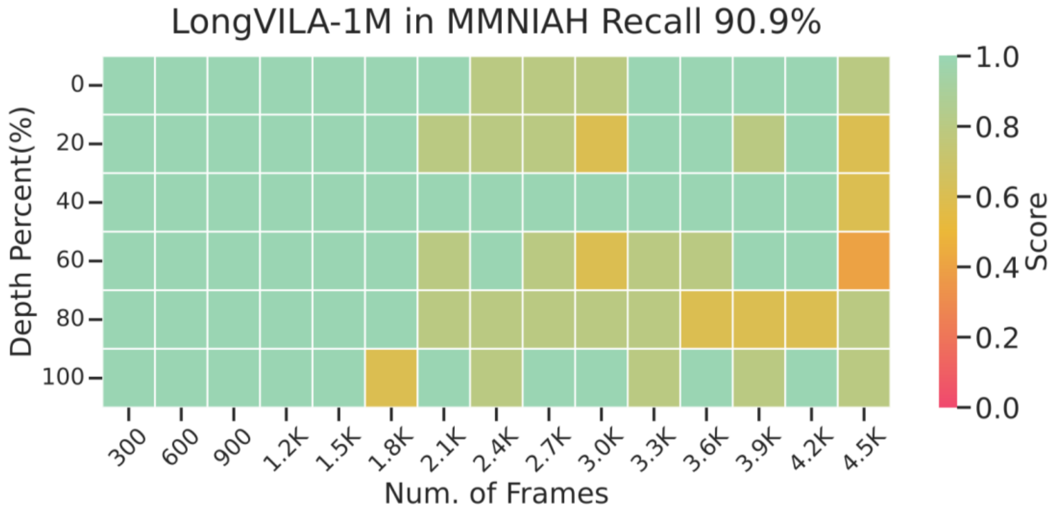
Table 1: Performance (%) of different models and different methods on video understanding tasks evaluated at frames from 110 to 256.

Model	FLOPs	VideoDC	ActNet-QA	EgoSchema	Next-QA	PerceptionTest	VideoMME		Avg.
		test	test	test	mc	val	wo/ 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**

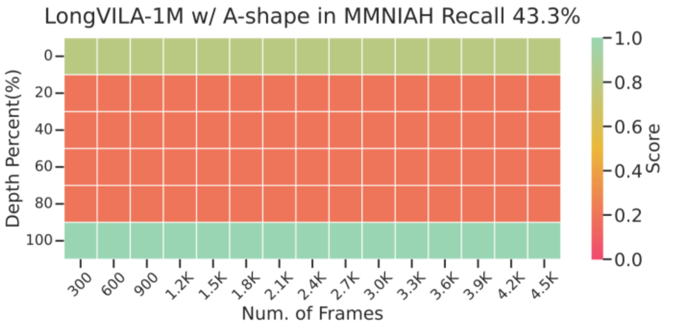
How effective is MMInference? V-NIAH



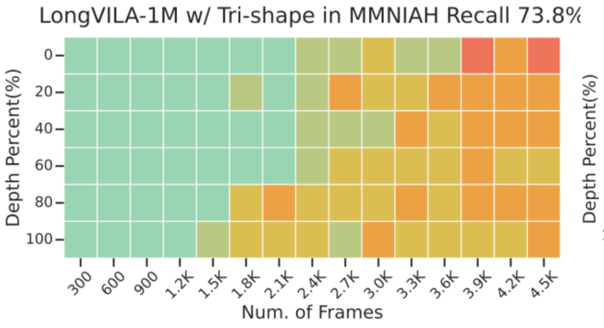
(c) MAPSparse in MM-NIAH



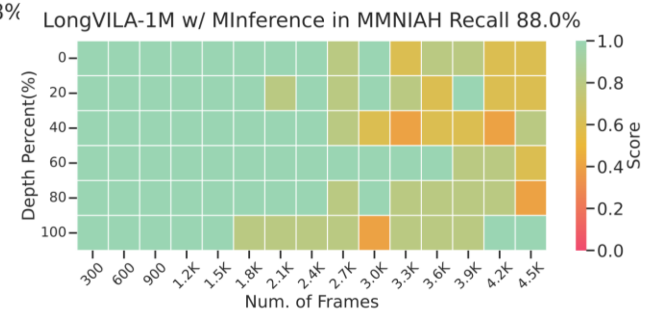
(d) FullAttention in MM-NIAH



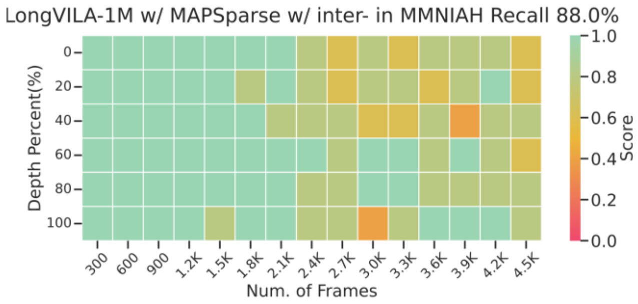
(a) A-shape



(b) Tri-shape

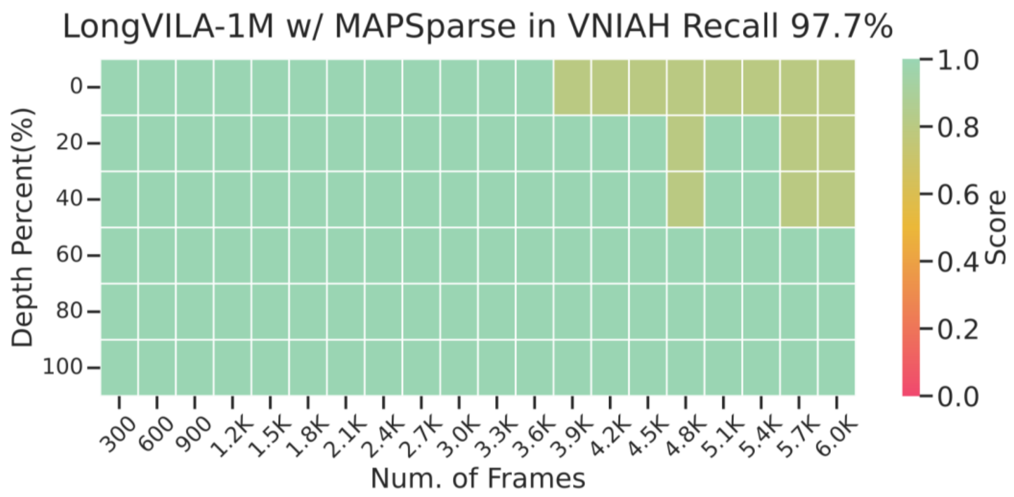


(c) MInference

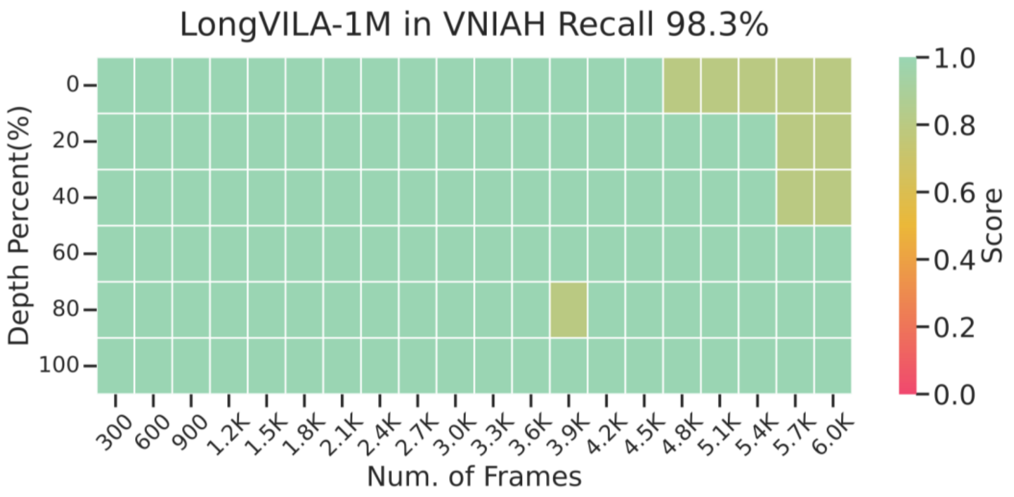


(d) MAPSparse w/ Inter-modality

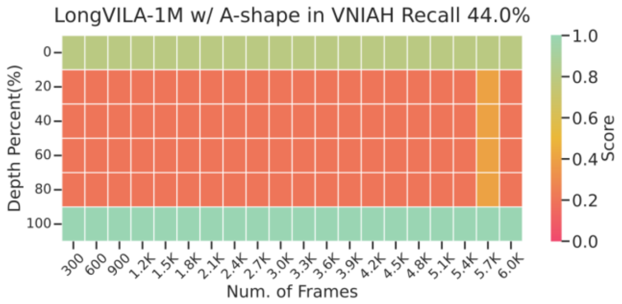
How effective is MMInference? MM-VIAH



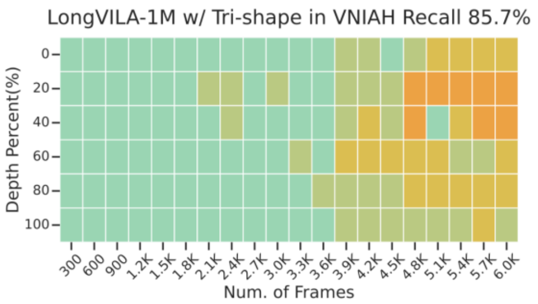
(a) MAPSparse in V-NIAH



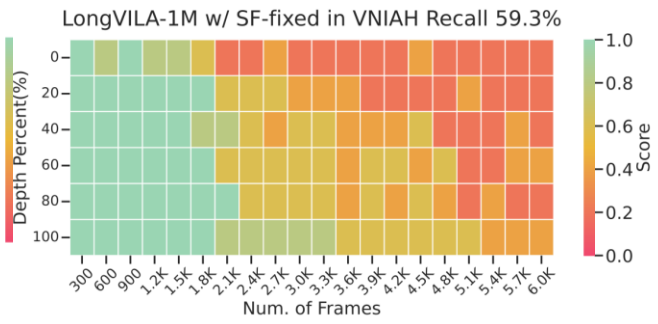
(b) FullAttention in V-NIAH



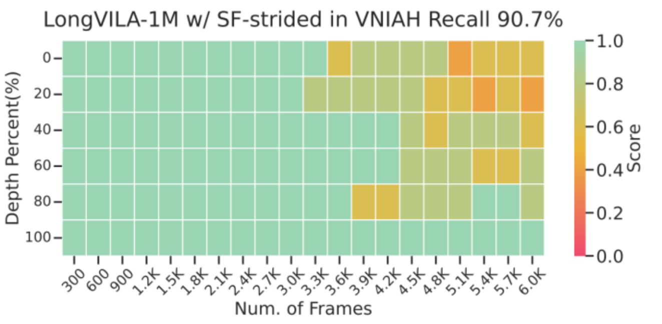
(a) A-shape



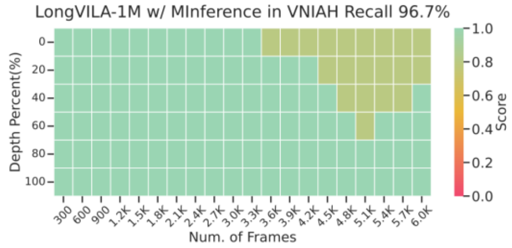
(b) Tri-shape



(c) SF-fixed

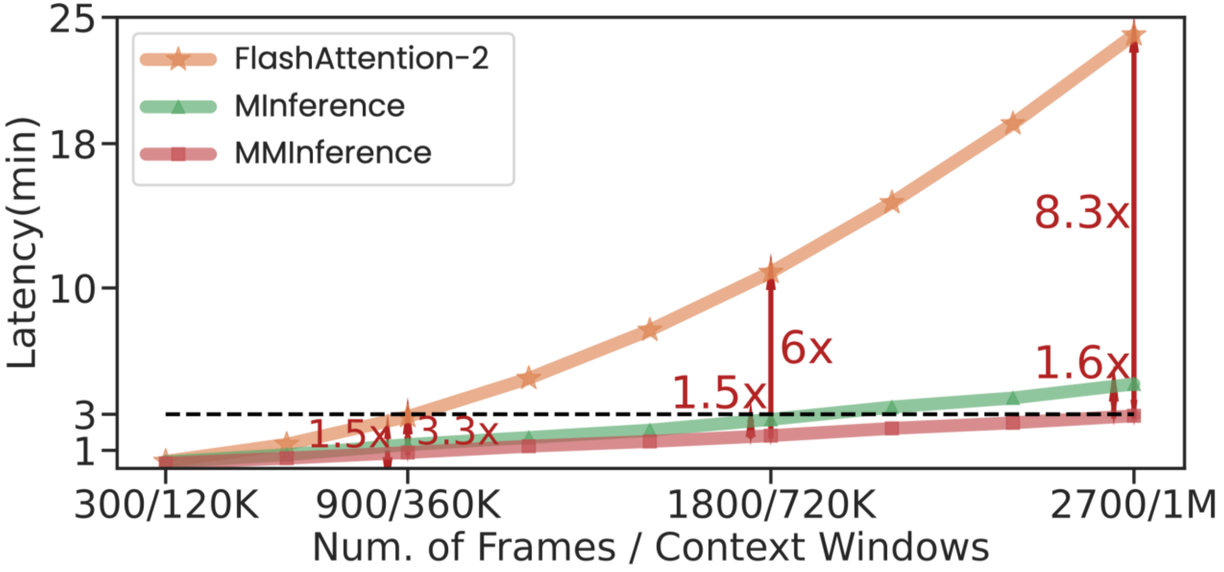


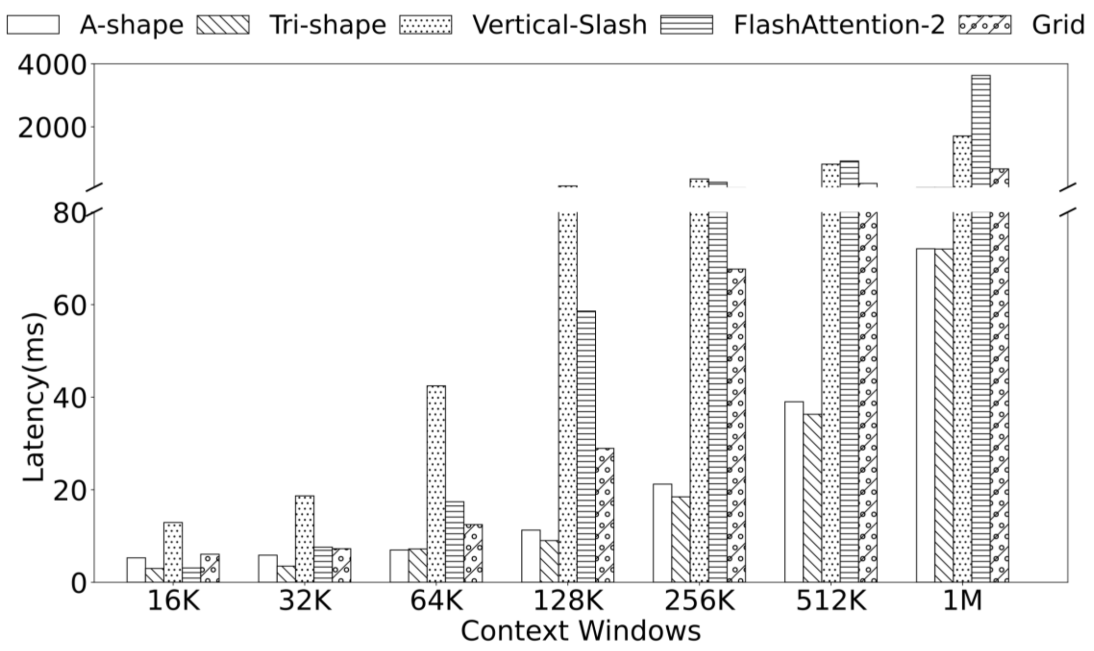
(d) SF-strided



(e) MInference

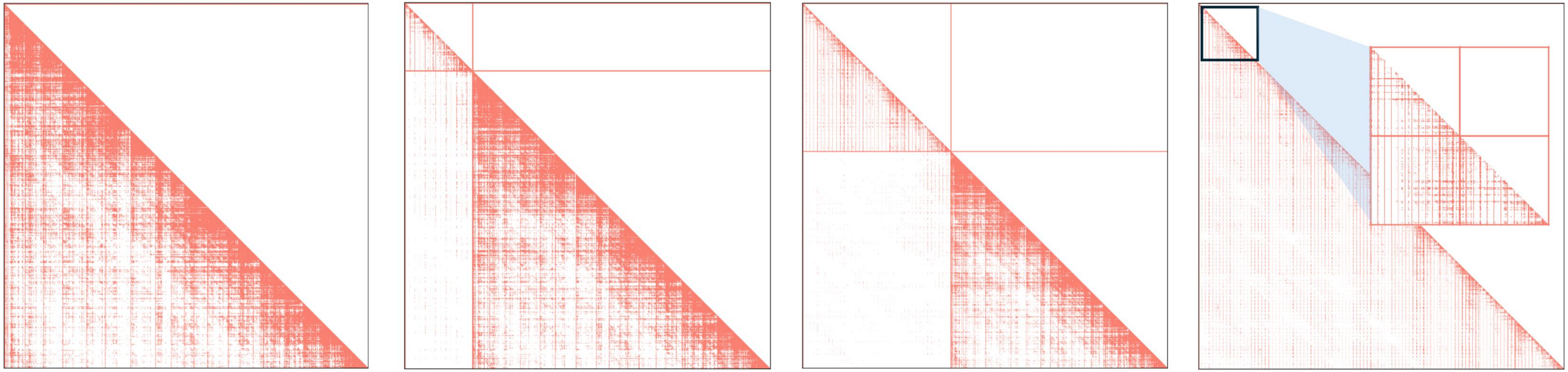
How efficient is MMInference? - E2E & MicroBench





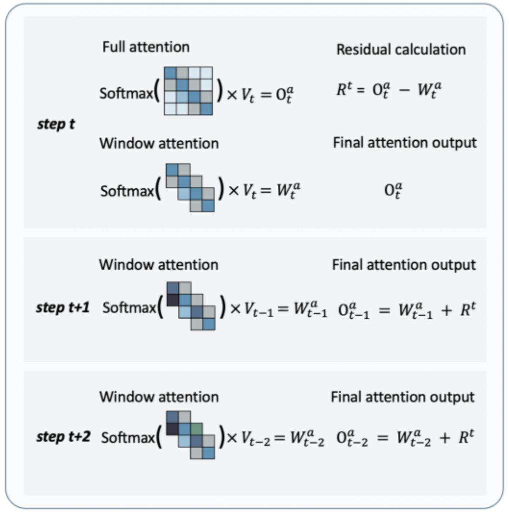
Transition of Sparse Patterns Across Modalities

- ❑ The VS pattern shifts to a Grid pattern when the input transitions from text to visual.



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

Discussion-Sparse DiT



DiTFastAttn

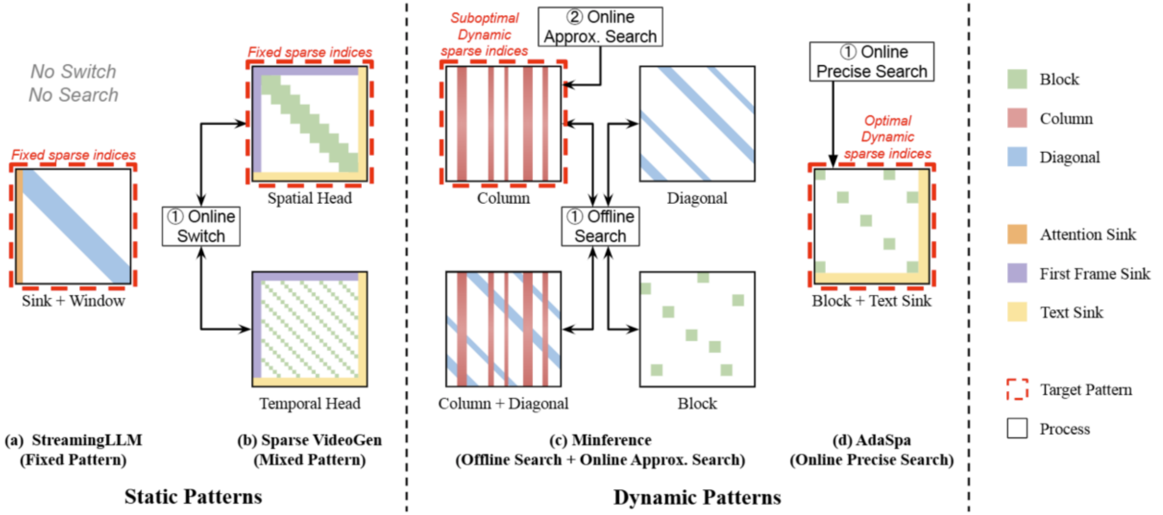
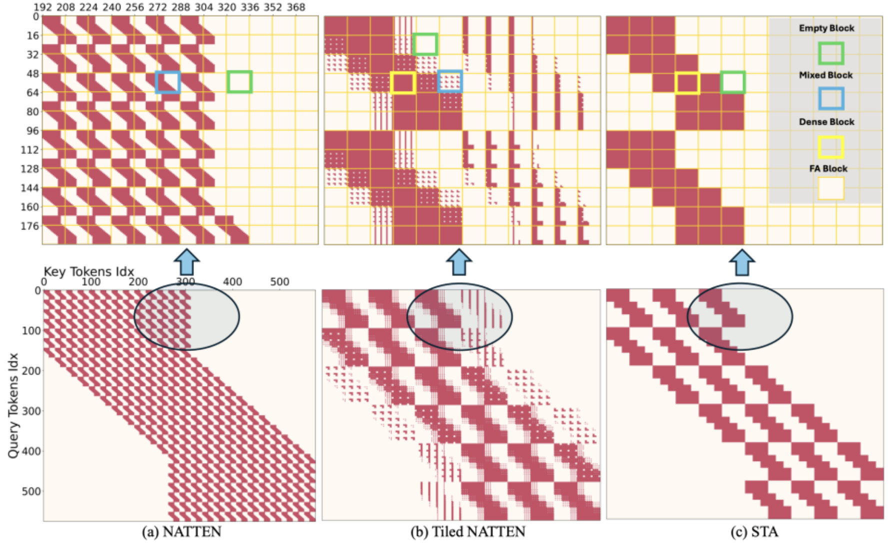


Figure 3. Different types of Sparse Pattern recognition methods. (a) StreamingLLM: using a static *sink+sliding window* pattern, need no search or switch. (b) Sparse VideoGen: preparing two predefined Static Patterns, and using an online switching method to determine which to use. (c) MInference: preparing several dynamic patterns, first do an offline search to determine the target pattern to use, then perform an online approximate search to search suboptimal sparse indices of this pattern. (d) AdaSpa: our method proves that the most suitable pattern for DiT is *blockified* pattern, and performs an online precise search to find the optimal sparse indices for blockified pattern.

Sparse VideoGen AdaSpa



STA

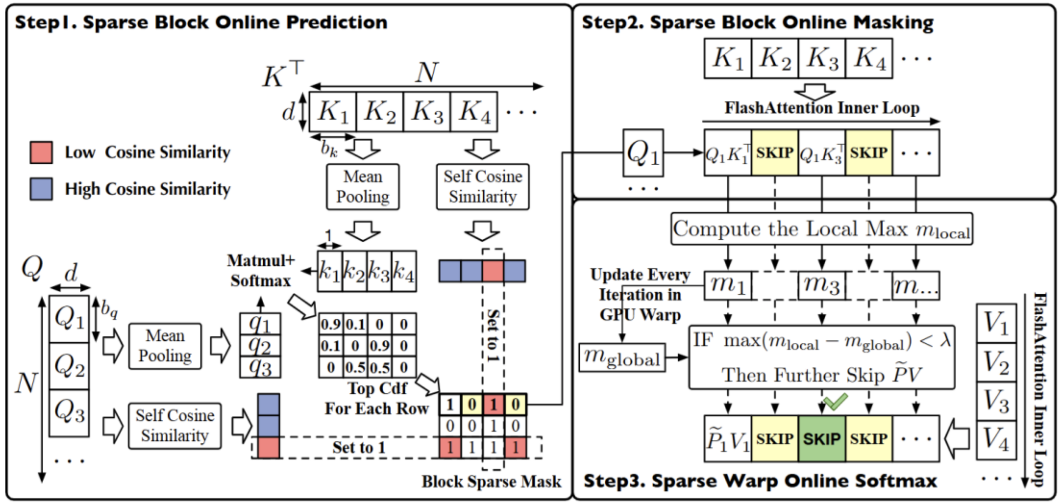


Figure 3. Workflow of `SpargeAttn`.

SpargeAttn