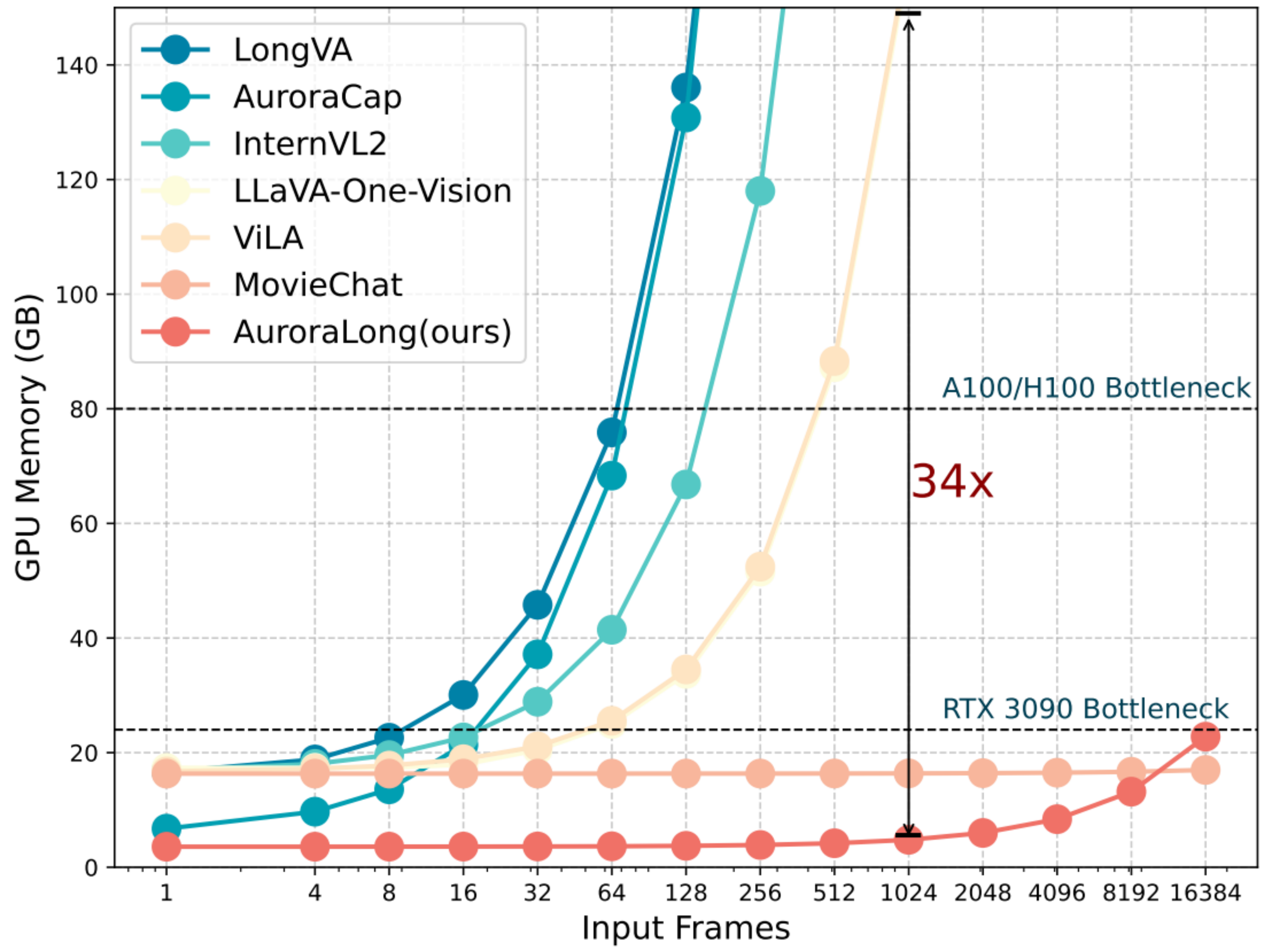
AuroraLong: Bringing RNNs Back to Efficient Open-Ended Video Understanding

Weili Xu[1,2,*] Enxin Song[1,*] Wenhao Chai[3,†] Xuexiang Wen[1] Tian Ye[4] Gaoang Wang[1,✉]
[1] Zhejiang University [2] University of Illinois Urbana-Champaign [3] Princeton University [4] Hong Kong University of Science and Technology (Guangzhou)
*Equal contribution, † Project lead, ✉ Corresponding Author.

ICCV HONOLULU HAWAII
OCT 19-23, 2025

Motivation & Contribution



Motivation

Video-based LMMs typically follow an architecture similar to LLaVA-NeXT. This approach has shown promising results but faces efficiency challenges when processing long videos with complex temporal dynamics. Recent studies demonstrate that increasing the number of sampled frames during training and inference substantially improves model performance. However, such improvement comes with considerable computational and memory costs. As the number of sampled frames increases, the computation overhead in transformer-based LLMs scales quadratically with number of input frames due to causal self-attention mechanism, where each token attends to all previous tokens.

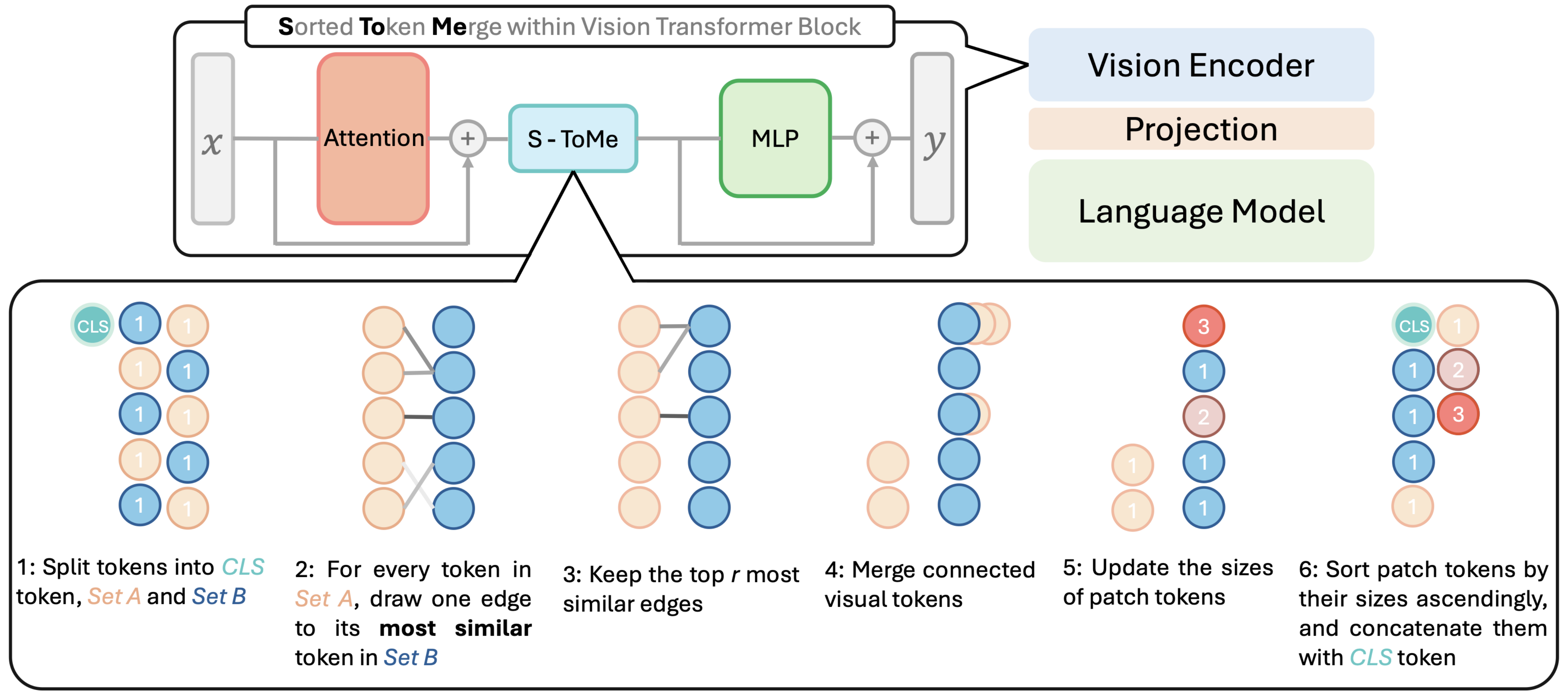
Contribution

🚀 We are the first to use a fully recurrent LLM backbone in a LLaVA-like model architecture for open-ended video QA, presenting a novel hybrid architecture that can handle long video inputs with lower memory requirement.

🚀 We propose a training-free Sorted Visual Token Merge strategy (S-ToMe) to increase model throughput while retaining spatial information for RNN-based large language models.

🚀 Despite only trained on public data, our model performs favorably against several state-of-the-art larger LMMs across various video understanding tasks, while reducing computational complexity and memory consumption.

Method



1: Split tokens into *CLS* token, *Set A* and *Set B*

2: For every token in *Set A*, draw one edge to its **most similar** token in *Set B*

3: Keep the top *r* most similar edges

4: Merge connected visual tokens

5: Update the sizes of patch tokens

6: Sort patch tokens by their sizes ascendingly, and concatenate them with *CLS* token

Figure 2. Visualization of the Sorted Token Merge (S-ToMe) algorithm used in AURORALONG, whose original version is in Appendix A.

Algorithm 1 Sorted Visual Token Merge

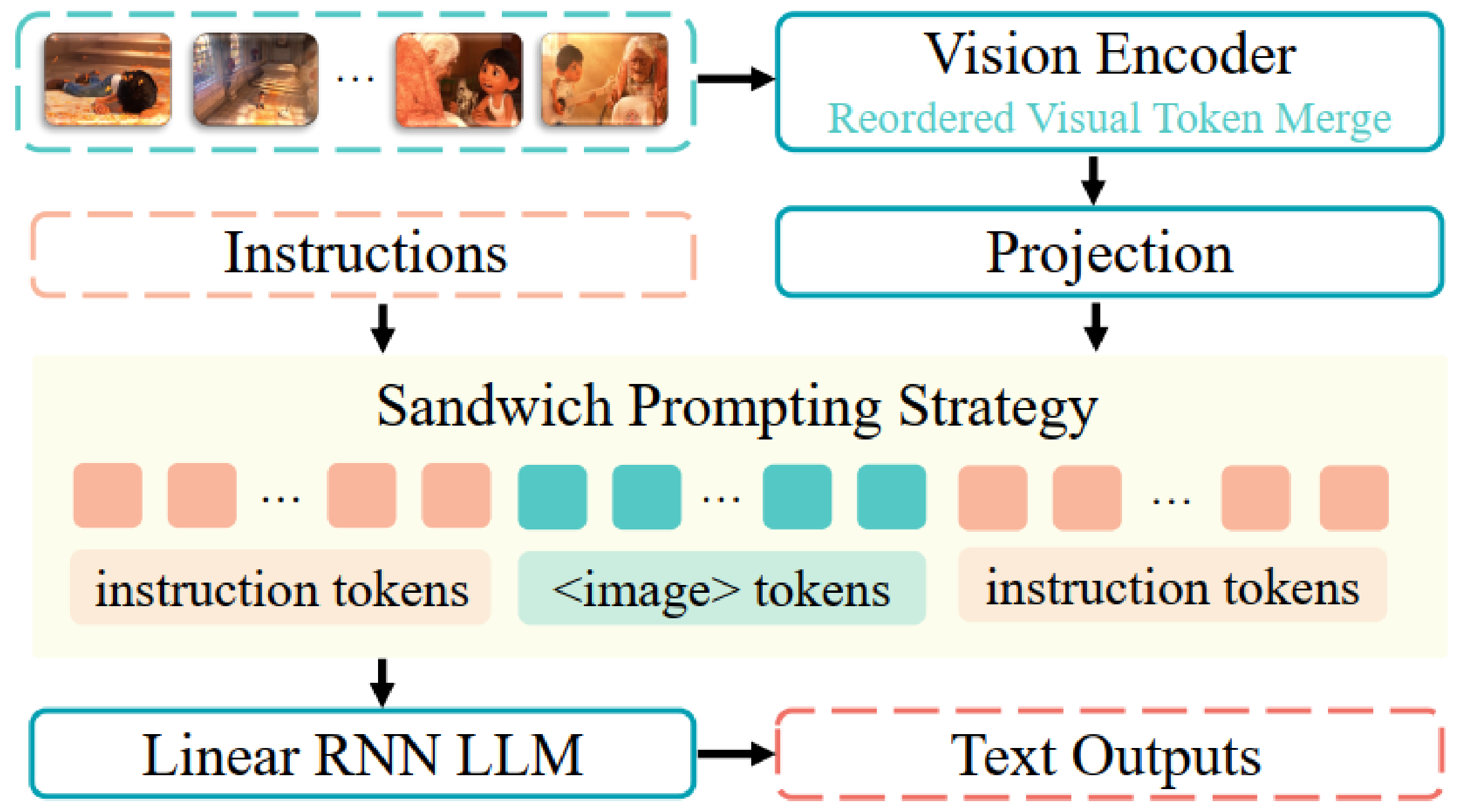
```
Require: Input visual tokens per frame $\mathcal{X}$
Require: Vision Transformer $\mathcal{V}$ with $\mathcal{N}$ layers
Require: Token Merging threshold $r$
  for $n$ in $\mathcal{V}[:\mathcal{N}-1)$ do
      # $\mathcal{X} \in [batch, tokens, channels]$
      $\mathcal{X} \leftarrow \text{Attention}_n(\mathcal{X})$
      # Split $CLS$ tokens and patch tokens
      $CLS, \mathcal{X} \leftarrow \mathcal{X}[:, 0, :], \mathcal{X}[:, 1 :, :]$
      # Assign patch tokens to $Set\ \mathcal{A}$, $Set\ \mathcal{B}$
      $\mathcal{A}, \mathcal{B} \leftarrow \mathcal{X}[:, :: 2, :], \mathcal{X}[:, 1 :: 2, :]$
      $Scores \leftarrow \text{similarity}(\mathcal{A}, \mathcal{B})$
      # Get merged tokens and unmerged tokens
      $src, unm \leftarrow \text{top}(\mathcal{X}, Scores, r)$
      $dst \leftarrow \text{merge}(src)$
      # Update patch count $s$ for each token
      $\text{update}(dst.s)$
      # Sort tokens by $s$
      $\mathcal{X} \leftarrow \text{sort}(dst, unm)$
      $\mathcal{X} \leftarrow \text{concat}(CLS, \mathcal{X})$
      $\mathcal{X} \leftarrow \text{MLP}(CLS, \mathcal{X})$
  end for
```

Core Design

🚀 To incorporate more frames within RWKV's limited pretrained context length, we propose Sorted Token Merge to reduce input sequence length to the LLM backbone while preserving spatial information.

🚀 To enhance AuroraLong's instruction following, we use the sandwich prompt, inserting the reordered merged visual tokens between the instruction tokens.

Prompting Strategy



Experiments

Short Video Question-Answering and Captioning

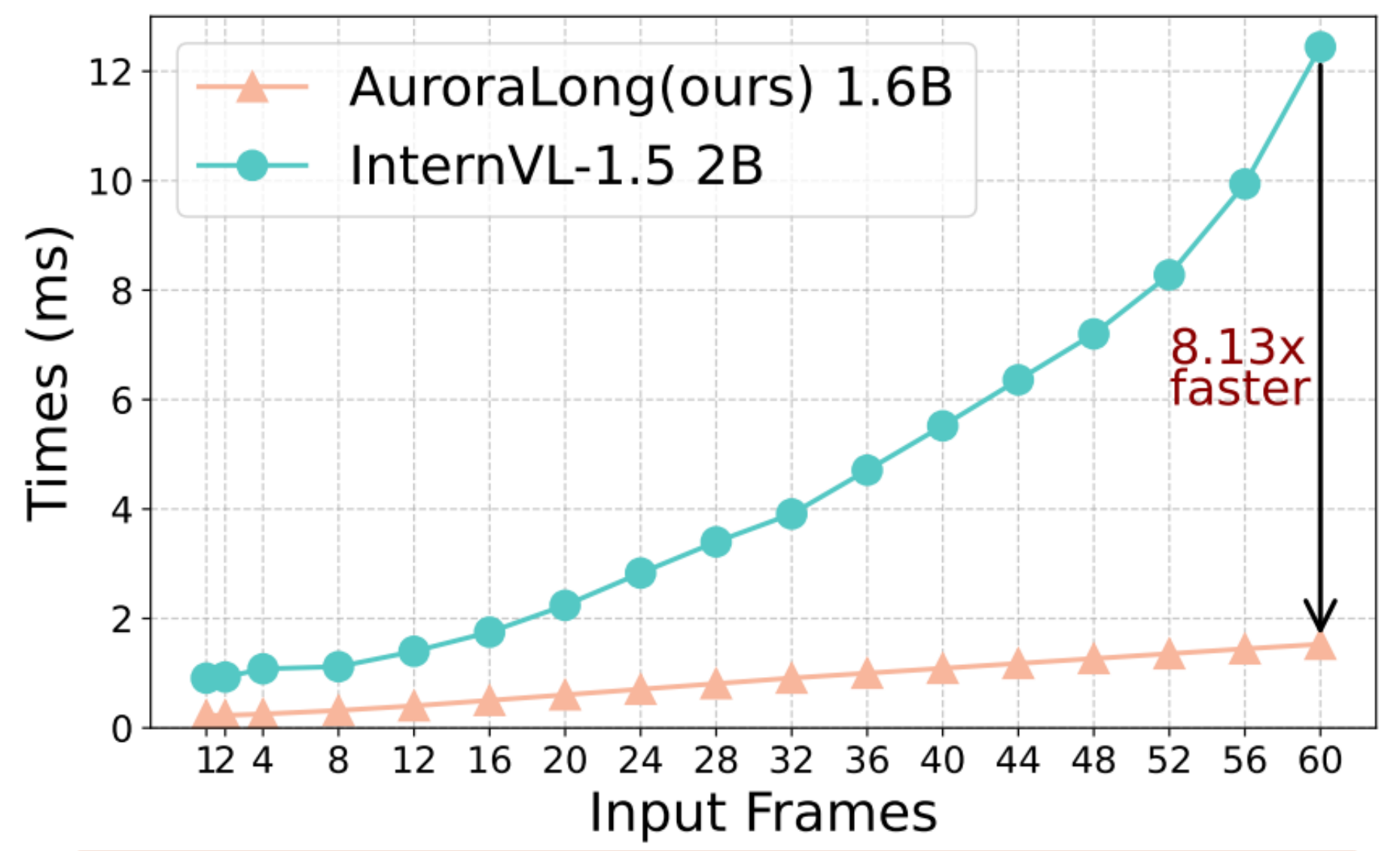
Models	Size	#Frame	VDC [9]						ANet [8]		VATEX [91]
			Avg.	Short	Camera	Background	Main Object	Detailed	Acc.	Score	BLEU@1
LongVA [101]	7B	64	34.50	31.94	35.32	36.39	40.95	27.91	-	2.8	65.2*
ShareGPT4Video [13]	8B	16	36.17	39.08	33.28	35.77	37.12	35.62	-	-	56.6*
LLAVA-OneVision [44]	7B	32	37.45	32.58	37.82	37.43	38.21	41.20	56.6	-	54.2*
AuroraCap [9]	7B	16	38.21	32.07	43.50	35.92	39.02	41.30	**61.8**	3.8	57.1
InternVL-2 [16]	8B	16	37.72	33.02	39.08	37.47	44.16	34.89	-	-	-
AuroraLong (ours)	2B	1fps	**42.54**	**38.89**	**43.70**	40.26	46.32	**43.54**	60.0	**4.2**	**68.5**

Long Video Question-Answering

Models	Input	CTX	Size	MLVU								MovieChat-1K	
				AVG	AR	ER	AO	AC	TR	NQA	PQA	Global	Break
GPT4-o	0.5fps	128k	-	**54.5**	**68.8**	47.8	46.2	35.0	**83.7**	42.9	57.1	-	-
LLaVA-OneVision* [44]	32 frm	132k	0.5B	50.3	58.5	52.4	28.6	30.9	67.0	33.3	42.8	-	-
Qwen2-VL* [90]	32 frm	132k	2B	48.7	54.7	47.6	30.9	28.6	73.8	40.4	60.5	-	-
InternVL2* [17]	32 frm	200k	2B	48.2	57.4	**57.1**	35.7	33.4	66.7	28.5	50.0	-	-
LongVA [101]	256 frm	224k	7B	42.1	41.0	39.6	17.1	23.3	81.3	46.7	46.0	55.9	56.5
ShareGPT4Video [13]	16 frm	8k	8B	34.2	25.6	45.3	17.1	8.3	73.6	31.7	38.0	69.0	60.9
InternVL-1.5 [18]	16 frm	8k	26B	37.9	51.3	24.5	14.3	13.3	80.2	40.0	42.0	57.7	61.1
VILA-1.5 [53]	14 frm	276k	40B	46.2	56.4	35.8	34.3	11.7	84.7	38.3	**62.0**	57.2	60.1
AuroraLong **(ours)**	48 frm	4k	2B	52.7	59.5	**57.1**	33.2	**42.9**	69.0	45.2	61.9	**84.0**	**64.0**

Models	Size	MVBench																			
		Avg.	UA	AC	MA	OE	ST	AL	AP	AS	CO	CI	EN	FGA	MC	MD	OI	OS	SC
LLaVA-OneVision [44]	0.5B	45.5	72.5	43.5	49.5	50.0	85.5	12.5	41.0	54.0	49.0	35.5	21.5	42.0	33.0	17.5	61.0	32.5	45.5
InternVL2* [17]	2B	52.9	60.5	30.5	**78.0**	79.0	83.5	31.0	**67.0**	**72.0**	36.0	**55.0**	32.0	38.0	**65.5**	32.0	64.0	30.0	44.5
Qwen2-VL* [90]	2B	**53.5**	73.0	43.5	75.5	**82.0**	82.0	12.5	41.0	54.0	49.0	35.5	21.5	**48.0**	55.0	45.0	55.0	29.5	43.0
LongVA* [101]	7B	50.8	68.5	47.0	56.5	49.5	89.0	45.0	58.0	55.6	61.5	41.0	**39.0**	43.5	28.0	36.5	**65.5**	30.5	49.0
ShareGPT4Video* [13]	8B	47.2	56.5	34.0	74.5	81.8	84.5	34.5	48.0	45.2	46.0	51.0	25.0	35.0	60.5	**54.0**	56.5	33.0	50.0
InternVL-1.5* [18]	26B	50.6	73.5	27.5	62.5	44.0	**89.5**	39.3	61.0	62.0	**64.0**	40.5	34.5	46.5	33.0	36.0	**65.5**	28.5	53.0
VILA-1.5* [53]	40B	42.7	60.0	41.5	34.5	50.0	69.5	36.5	39.5	40.5	44.0	40.0	27.0	33.0	37.0	27.5	59.5	38.0	47.5
AuroraLong **(ours)**	2B	53.2	**75.0**	**52.0**	65.5	62.5	87.0	**48.0**	47.5	49.5	47.0	52.0	35.0	46.5	48.5	44.0	54.0	37.5	**53.5**

Efficiency Analysis



When compared to **InternVL-1.5 2B**, a transformer-based model of similar size, **AuroraLong** requires less computation and provides an 8.13×lower latency when handling a video input with 60 frames.

Ablation Study

Token Order	ANet [45]	VATEX [42]	VDC [5]	MovieChat-1K [36]
Random	53.1	67.6	40.9	76.5
Descending	55.0	67.0	41.1	76.0
Ascending	**56.3**	**68.5**	**41.3**	**78.5**

We conduct careful ablation on input order for merged visual tokens within a frame, and observe that ascending token merging performs best, likely because larger patches contain critical information for tasks like visual question answering, making it easier for RWKV to utilize its data-dependent token shifting mechanism and memorize the most critical information for visual question-answering of each frame.