Adaptive Keyframe Sampling for Long Video Understanding
-supplementary material-

Xi Tang[1]*, Jihao Qiu[1]*, Lingxi Xie, Yunjie Tian[2], Jianbin Jiao[1], Qixiang Ye[1]
[1]University of Chinese Academy of Sciences
[2]University at Buffalo, SUNY
{tangxi19,qiujihao19}@mails.ucas.ac.cn {jiaojb,qxye}@ucas.ac.cn

A. Comparsion with Other Long Video Understanding Methods

We first compare the accuracy of video question answering between our approach and the methods extended the context length of language model(*e.g.*, LongVA [11] and LongVILA [10]) or utilized token reduction(*e.g.*, CrossGET [4], VCWE [8] and LongVU [3]) to allow more frames to be encoded and processed. As in Table 1, our method can achieve better results with the same even fewer frames, since they do not address the issue of irrelevant or redundant information among frames while our method focuses on selecting keyframes to ensure that the input contains relevant information. Additionally, our methods can also be integrated with them (*e.g.* LongVA [11], CrossGET [4] and LongVU [3]) for improvement as showen in Table 1.

We also compare the accuracy of video question answering between our approach and the methods (*e.g.*, VideoAgent [6] and VideoTree [7]) selected keyframes using language (from the LLMs). As in Table 1, our method exceeds their performance with same even fewer frames, since these methods selected keyframes solely using language, which can miss important visual cues as shown in Figure 1.

B. Details of the ADA Algorithm

In Algorithm 1, we present the detailed pseudocode of our **ADA** algorithm. To accelerate the experimental process, we pre-process the video frames (sampled at 1 frame per second along with the corresponding questions) by inputting them into the VL scorer to obtain the corresponding scores.

Algorithm 1 ADA: Adaptive Keyframe Selection

Input: $matching\_score$: a list, where each element is the matching score of a frame and the corresponding question
$level$: Current recursion level
max_level: maximum recursion level
s_{thr}: threshold
M: number of frames to select
Output: selected_frames: Indices of the selected M frames

Function `SplitSegments`(matching_scores, $level$, max_level,s_{thr}, M) **:**
- split_scores ← []
  `// List of completed segments`
- new_scores ← []
  `// List of segments to further split`
- **foreach** $matching\_score$ *in* matching_scores **do**
  - s_{all} ← mean(matching_score)
  - s_{top} ← mean(topk(matching_score, M))
  - m← $s_{\text{top}} - s_{\text{all}}$
  - **if** $m \geq s_{\text{thr}}$ **then**
    - Append matching_score to split_scores
  - **else if** $level < max\_level$ **then**
    - Split $matching\_score$ into two bins from the center, denoted as $split1$ and $split2$
    - Append $split1$ and $split2$ to new_scores
- **if** *new_scores is not empty* **then**
  - deeper_scores ← `SplitSegments` (new_scores, $level + 1$, max_level,s_{thr}, $M//2^{level}$)
  - split_scores ← merge(split_scores, deeper_scores)
- **return** split_scores

*Equal contribution.

Function `SelectFrames` (*segments*, M) **:**
  total_length ← total length of all *segments*
  selected_frames ← []
  foreach *segment in segments* **do**
    $m_i \leftarrow \lfloor M \times \text{length}(segment)/\text{total\_length} \rfloor$
    Select the top m_i highest-scoring frame indices from *segment*
  Append the selected indices to selected_frames
  return selected_frames

Main:
  matching_scores ← [*matching_score*]
  segments ← `SplitSegments` (matching_scores, *level*, *max_level*,s_{thr})
  selected_frames ← `SelectFrames` (*segments*, M)
  return selected_frames

These scores are then stored in a list referred to as *matching_score*. Each element in *matching_score* consists of the matching score for a specific video frame and the corresponding question. We begin by employing a recursive strategy to partition the matching_scores list into sublists of varying lengths, according to the partitioning rule outlined in Section 3.3. Subsequently, based on the lengths of these sublists, we select different numbers of frames with the highest matching scores from each sublist to construct the final set of video frames. This final set is then sent to the language model for visual understanding.

C. More Visualization Results

In Figure 2, we show more examples of video understanding results of AKS (based on three baselines, LLaVA-Video-7B [12], Qwen2-VL-7B [5], and LLaVA-OV-7B [2]). As shown, our approach benefits from the ability to locate keyframes so that the MLLM receives effective visual information for understanding. The ability easily transfers to various MLLMs in a plug-and-play manner.

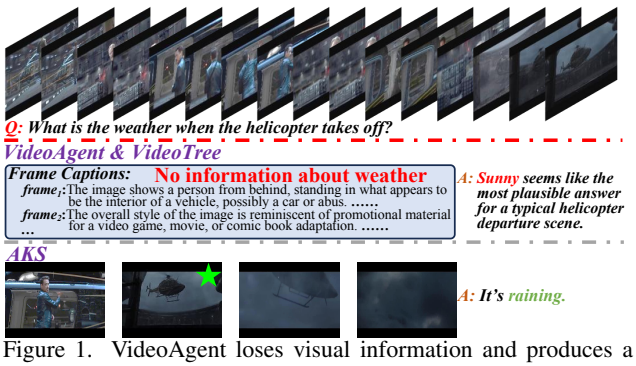

Figure 1. VideoAgent loses visual information and produces a wrong answer. AKS looks into the keyframe for the right answer.

Table 1. Comparsion with Other Methods of Dealing with Long Videos

Method	Frames	LVB [9]	V-MME [1]
LongVA	64	52.8	52.4
LongVA + AKS	64	**55.4**	**53.6**
LongVILA	256	57.1	60.1
LLaVA-Video + AKS	64	**62.7**	**65.3**
LLaVA1.5	6	42.1	39.3
+CrossGET	6	41.2	40.1
+ AKS	6	**43.9**	**40.9**
+ CrossGET & AKS	6	**44.1**	**41.8**
VCWE	512	58.0	60.2
LongVU	1fps	52.4	55.0
LongVU + AKS	64	**55.1**	**56.1**
LLaVA-Video + AKS	64	**62.7**	**65.3**
VideoAgent	5 (avg)	-	43.3
LLaVA-Video + AKS	4	-	**55.4**
VideoTree	-	52.8	52.4
LLaVA-Video + AKS	8	**57.8**	**60.0**

References

[1] 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 arXiv:2405.21075*, 2024. 2, 3

[2] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[3] 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 arXiv:2410.17434*, 2024. 1

[4] Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *arXiv preprint arXiv:2305.17455*, 2023. 1

[5] 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 arXiv:2409.12191*, 2024. 2

[6] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2025. 1

[7] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024. 1

[8] Hongchen Wei and Zhenzhong Chen. Visual context window extension: A new perspective for long video understanding. *arXiv preprint arXiv:2409.20018*, 2024. 1

[9] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li.

Figure 2. More examples of AKS enhance the baseline MLLMs for video understanding. The left three examples come from LongVideoBench [9] while the right three are from VideoMME [1]. Green stars indicate keyframes selected by AKS.

Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024. 2, 3

[10] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 1

[11] 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 arXiv:2406.16852*, 2024. 1

[12] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 2