Sharper and Faster mean Better: Towards More Efficient Vision-Language Model for Hour-scale Long Video Understanding

Daoze Zhang, Yuze Zhao, Jintao Huang, Yingda Chen
Alibaba Group
{zhangdaoze.zdz,yuze.zyz,huangjintao.hjt,yingda.chen}@alibaba-inc.com

Abstract

Despite existing multimodal language models showing impressive performance on the video understanding task, extremely long videos still pose significant challenges to language model's context length, memory consumption, and computational complexity. To address these issues, we propose a vision-language model named Sophia for long video understanding, which can efficiently handle hour-scale long videos. First, we employ a Shot-adaptive Frame Pruning technique, which naturally segments long videos into multiple camera shots, to *more sharply* identify and focus on the frames relevant to the query. Additionally, we introduce a Hierarchical Attention mechanism to effectively model the long-term temporal dependencies between video frames, which achieves a time and space complexity of $\mathcal{O}(N)$ w.r.t. the input sequence length N while theoretically maintaining the global modeling efficiency. Experimentally, our Sophia exhibits competitive performance compared to existing video understanding baselines across various benchmarks for long video understanding with reduced time and memory consumption. The model code and weights are available at this repository.

1 Introduction

Fueled by the rapid advancements of large language models (LLMs) (Achiam et al., 2023; Dubey et al., 2024), multimodal large language models (MLLMs) have undergone remarkable development, especially in vision-language models (VLMs) (Li et al., 2023; Liu et al., 2024a; Wang et al., 2024a). These VLMs evolved from single-image analysis to handling multiple images, and now advanced to understanding videos, which contain latent causal relationships between frames. In the realm of video understanding or question answering (QA), researchers commonly sample frames uniformly from a video, thereby recasting the problem as a task to understand multi-image sequences with temporal dependencies. Although this straightforward idea may work well in many short video scenarios (Zhang et al., 2023; Maaz et al., 2023; Papalampidi et al., 2024), when it comes to long videos from ten minutes to an hour, the sheer volume of visual tokens—reaching tens of thousands—poses considerable challenges to the LLM's context length, memory consumption, and computational complexity.

To address the challenge of excessively long visual token sequences, existing research on long video understanding primarily diverges into two main work lines. Some researchers concentrate on minimizing the number of tokens of each frame (Wang et al., 2024c; Shu et al., 2024; Chen et al., 2024b) and adopt memory banks (Song et al., 2024; He et al., 2024) to limit the length of visual token sequences, while maintaining the sampling rate of frames. However, this obviously compromises the model's ability to capture details from the long video. Another line of research (Wang et al., 2024d; Shen et al., 2024; Ataallah et al., 2024; Wang et al., 2024e) insightfully notices that processing all information in long videos is both impractical and unwise due to the significant noise and redundancy inherent in long videos. Consequently, most of these methods focus on dropping frames that are irrelevant to specific queries. However, the majority of these works segment videos into fixed-length clips and discard the clips deemed unrelated, ignoring the dynamic and temporally uneven nature of events or camera shots in long videos, resulting in inaccurate frame pruning.

To preserve detailed video information while avoiding inaccurate frame dropping, we propose a novel two-stage Shot-adaptive Frame Pruning technique, which segments videos into camera shots or events naturally based on the semantics of frames rather than equal splitting. In this way, the model can adaptively learn to figure out the scenes that are most relevant to the query, thereby more sharply

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4423–4439
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

identifying the frames of interest. Specifically, a specialized shot detection module is employed to divide the video into events or scenes based on the semantic differences between consecutive frames. Then, at the *Inter-shot Pruning* stage, we first remove shots that are irrelevant to the query, to prune the obvious noise in long videos from a coarse level. After that, at the *Intra-shot Pruning* stage, noting that there commonly exists semantical redundancy within multiple consecutive frames of a single shot, we perform frame selection within each shot based on the similarity between frames.

In addition, from a lower-level perspective, inspired by the development of sparse attention mechanisms (Beltagy et al., 2020; Chalkidis et al., 2022; Hatamizadeh et al., 2023), the potential of sparse attention on long video token sequences warrants further exploration. However, while sparse attention techniques reduce the computational complexity, most of them may sacrifice the efficiency with which distant tokens access global information. For instance, if we directly use (dilated) sliding window attention with window size w, the two frames with a spacing of F frames require an information propagation path of length $\lceil F/w \rceil = \mathcal{O}(F)$ to receive information from each other, whereas vanilla dense attention achieves this with a path of length 1. In this paper, we use the term "*information propagation distance (IPD)*" to describe the length of the shortest information propagation path between two frames, which equals to the number of stacked attention layers necessary for them to share information. For a certain sparse attention mechanism, a lower IPD between temporally distant frames is preferable (with vanilla attention keeping IPD $= 1$), which measures the global modeling efficiency of the sparsification method. As mentioned above, the high cost in IPD of most existing sparse attention hinders the effective modeling of global information in long videos.

To benefit from low-complexity sparse attention while achieving a low IPD, we introduce a Hierarchical Attention mechanism that aggregates information from consecutive frames at multiple granularities, such that LLMs can efficiently process long visual token sequences. It can be proved that the Hierarchical Attention not only reduces the time and space complexity of vanilla attention from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ w.r.t. the input token number N, but also can maintain an IPD of $\mathcal{O}(1)$, which is close to that of vanilla attention. This method not only addresses the high memory consumption and computational costs in long video understanding from a lower level, but also theoretically ensures that the capabilities of LLMs will not be greatly affected by the sparsification.

Combining the above-mentioned **Sho**t-adaptive Frame **P**runing and **Hi**erarchical **A**ttention, we propose a vision-language model, Sophia, for long video understanding that can handle hour-scale long videos. Overall, our contributions comprise:

- To our knowledge, we are the first to propose a shot-aware method that segments videos naturally based on camera scenes, which can more sharply identify the frames relevant to a specific query, to prune the noise and redundancy from the outset for long video understanding.
- From a lower level, we employ the Hierarchical Attention in the LLM, which achieves $\mathcal{O}(N)$ time and space complexity while maintaining global modeling efficiency with the same complexity as full attention (measured by IPD). This not only addresses the high costs of computation and memory, but also theoretically ensures the LLM's ability will not be greatly affected.
- Extensive experiments demonstrate that Sophia achieves the best performance on 6 out of 8 long video understanding datasets or benchmarks, showing competitive capabilities against SOTA baselines with less time and memory resources.

2 Method

The overall architecture of Sophia is shown in Fig. 1. First, we employ a lightweight shot detector that segments the video into shots based on the semantic information of frames, which allows the model to comprehend temporally uneven events or scenes in long videos more naturally. Then, each frame is fed into a vision encoder and a projector to obtain a set of visual tokens. During the Inter-shot Pruning stage, we assess the correlation between the visual embeddings of each shot and the textual embeddings of the user's query to prune the shots unrelated to the query. Then, considering that shots often contain continuous actions or identical scenes, the Intra-shot Filtering stage removes redundant frames within a shot. In this way, the model can adaptively learn to select the most attention-worthy frames (Sec. 2.1). Moreover, from a foundational level, we employ the sparse Hierarchical Attention in our LLM, to perform attention with $\mathcal{O}(N)$ complexity among video tokens while preserving competitive model capabilities (Sec. 2.2).

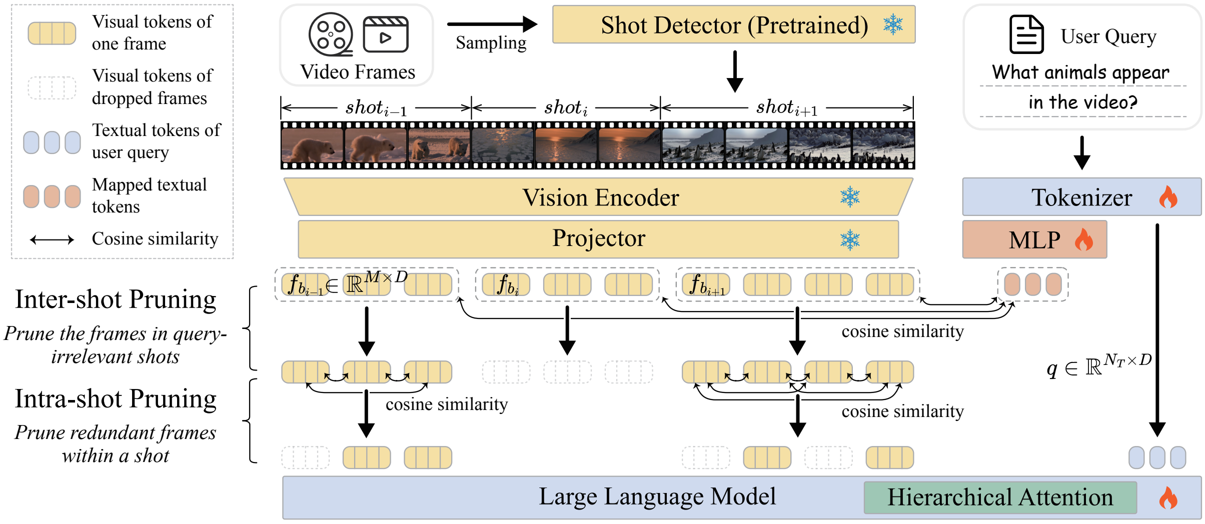
Visual tokens of one frame
Visual tokens of dropped frames
Textual tokens of user query
Mapped textual tokens
Cosine similarity
Video Frames
Sampling
Shot Detector (Pretrained)
User Query
What animals appear in the video?
$shot_{i-1}$
$shot_i$
$shot_{i+1}$
Vision Encoder
Projector
Tokenizer
MLP
$f_{b_{i-1}} \in \mathbb{R}^{M \times D}$
f_{b_i}
$f_{b_{i+1}}$
Inter-shot Pruning
Prune the frames in query-irrelevant shots
cosine similarity
$q \in \mathbb{R}^{N_T \times D}$
Intra-shot Pruning
Prune redundant frames within a shot
cosine similarity
cosine similarity
Large Language Model
Hierarchical Attention

Figure 1: Overview of Sophia. First, we employ a shot detector to segment the video into camera shots based on the semantics of sampled frames. Next, all frames are fed into a vision encoder and a projector to obtain visual tokens. To prune the noise and redundancy in long videos, we propose a two-stage frame pruning method: (1) the model first sharply identifies and keeps the query-relevant shots; and then (2) removes the redundant frames within each shot (Sec. 2.1). Finally, the visual tokens of the retained attention-worthy frames, along with the textual tokens, are fed into the LLM, which integrates the cost-performance-balanced sparse Hierarchical Attention mechanism (Sec. 2.2).

2.1 Shot-adaptive Frame Pruning

Shot Detection. To improve the limitation of existing works on equal-length video splitting, we propose to preprocess all sampled frames with a lightweight shot detector, allowing our model to adapt to temporally uneven events or scenes in the video. Specifically, we utilize the pre-trained TransNet (Soucek and Lokoc, 2020), a strong model on the transition detection task, as our shot detector. Formally, a frame sequence can be denoted as $\{r_n\}_{n=0}^{F-1}$, $r_n \in \mathbb{R}^{C \times H \times W}$, where F denotes the number of video frames, C denotes the number of image channels, and H and W denote the height and width of one frame, respectively. A frame sequence will be fed into the shot detector, yielding the shot segmentation of the video: $\{[r_{b_0}, r_{b_1 - 1}], [r_{b_1}, r_{b_2 - 1}], \ldots, [r_{b_{s-1}}, r_{F-1}]\}$, where $b_0 = 0, b_1, b_2, \ldots, b_{s-1}$ denote the indices of the beginning frames of the s detected shots.

Inter-shot Frame Pruning. At a coarse granularity, to prune the shots that are irrelevant to the query, we calculate the correlation between each shot and the text query. Specifically, all the frames $\{r_n\}_{n=0}^{F-1}$ are fed into a vision encoder and a projector to obtain the visual embeddings $\{f_n\}_{n=0}^{F-1}$, $f_n \in \mathbb{R}^{M \times D}$, where M denotes the number of tokens of one frame and D denotes the dimension of embeddings. To calculate the similarity, the embedding of the middle frame $f_{\lfloor (b_i + b_{i+1} - 1)/2 \rfloor}$ of the shot $[r_{b_i}, r_{b_{i+1} - 1}]$ is taken as the representative of the shot's semantic information. For the query embedding $q \in \mathbb{R}^{N_T \times D}$ where N_T denotes the text query sequence length, it will pass through a learnable MLP, and then calculate the cosine similarity $\mathcal{S}_{\text{inter}}$ with the semantic embedding of each shot (Formula 1). Specifically, here we first calculate the cosine similarity, then apply the pooling operation to obtain the final similarity between one query and one representative frame (but not pooling then calculating similarity). Among all the candidate shots, the irrelevant $\alpha\%$ of shots with the smallest $\mathcal{S}_{\text{inter}}$ values will be pruned.

$$\mathcal{S}_{\text{inter}}(f_n) = \text{cos-sim}(f_{\lfloor \frac{b_i + b_{i+1} - 1}{2} \rfloor}, \text{MLP}(q)) \quad (1)$$

Intra-shot Frame Pruning. After pruning irrelevant shots, there may still exist redundant frames within a shot, as consecutive frames often capture continuous actions and similar semantics within the same shot. For instance, the first two frames of the example video in Fig. 1 both depict the same polar bear running. To address this, for the shot $[r_{b_i}, r_{b_{i+1} - 1}]$ retained in the previous stage, we calculate the cosine similarity $\mathcal{S}_{\text{intra}}$ between each frame's embedding f_n and those of other frames as a score for keeping frame r_n (Formula 2). Specifically, given two frame representations $f_i, f_j \in \mathbb{R}^{M \times D}$, we first compute the similarity between their corresponding visual tokens, resulting in a similarity matrix of shape $M \times M$. We then take the mean value across both dimensions of this matrix to obtain the final similarity as a scalar. A redundancy rate of $\beta\%$ is pruned here.

$$\mathcal{S}_{\text{intra}}(f_n) = \sum_{n'=b_i, n' \neq n}^{b_{i+1}-1} \frac{\text{cos-sim}(f_n, f_{n'})}{b_{i+1} - b_i - 1} \tag{2}$$

Note that the domains of both Formula 1 and 2 are $b_i \leq n \leq b_{i+1} - 1, 0 \leq i \leq s - 1$.

In the training of frame pruning, direct indexing will prevent the back-propagation of gradients. Therefore, we use the differentiable Gumbel Softmax technique (Jang et al., 2017) in our experiments to equivalently complete the training of pruning. Through our two-stage frame pruning approach, the model can adaptively learn to select the frames of interest, which effectively eases the challenge of noise and redundancy in long videos.

2.2 Hierarchical Attention

From a lower-level perspective, we introduce the Hierarchical Attention (Fig. 2) in our LLM instead of vanilla dense attention, to reduce the quadratic time and space complexity to $\mathcal{O}(N)$ while maintaining an IPD of $\mathcal{O}(1)$. Next, we elaborate on the motivation and details of the Hierarchical Attention, including a theoretical proof of its efficiency.

Premise. In the long video understanding or question answering task, according to the common practice (Chen et al., 2024d; Wang et al., 2024c; Chen et al., 2024c), tens to thousands of frames are typically sampled from videos, each represented by tens to hundreds of tokens. In contrast, text queries contain merely a handful to a few dozen tokens. Therefore, it is obvious that the video tokens overwhelmingly dominate the input sequence length, while the text part is significantly shorter. Moreover, it is worth noting that for the long video issue we focus on, the length of the text query may not increase with the growth of video length. Formally, the entire long input sequence has a length of N, comprising a visual sequence of length N_V and $N_T = N - N_V \ll N_V$ textual tokens. Here N_T can be regarded as a constant. Then, the complexity of the full attention can be transformed as:

$$\begin{aligned} \mathcal{O}(N^2) &= \mathcal{O}((N_V + N_T)^2) \\ &= \mathcal{O}(N_V^2) + \mathcal{O}(N_V N_T) + \mathcal{O}(N_T^2) \\ &= \mathcal{O}(N_V^2) + \mathcal{O}(N_V). \end{aligned} \tag{3}$$

Noting the above fact, our Hierarchical Attention mainly focuses on the sparsification between video tokens, that is, it reduces the $\mathcal{O}(N_V^2)$ term to $\mathcal{O}(N_V)$ in Formula 3 (details are below).

Sparse Attention among Frames. The motivation of our Hierarchical Attention is two-fold. Initially, to fully exchange information between frames, the vanilla dense attention needs performing attention operations between any two frames, resulting in a $\mathcal{O}(N_V^2)$ complexity. To improve this, we seek to limit the number of frames that each frame needs to perform attention with, avoiding the necessity of attending to all other frames. To achieve this, we adopt the hierarchical idea to create *coarse-grained* frames by aggregation, which has been utilized in pure text and single image scenarios (Chalkidis et al., 2022; Hatamizadeh et al., 2023). Specifically, as shown in Fig. 2, firstly the visual tokens of adjacent A frames are hierarchically aggregated using a learnable CNN in a non-overlapping manner, forming an A-ary tree structure with the original frames as leaf nodes. We use $0 \leq l \leq L - 1$ to denote the level index from bottom to top, where $L - 1$ is the number of aggregation operations, a constant (so it may form a trapezoid but not a strict tree for small A and L).

By hierarchically aggregating the original visual tokens, each frame can focus on a coarser frame at a higher level, to capture the semantics of all descendants of the coarser frame through stacked attention layers. To achieve this, attention operations are supposed to be performed between all parent-child frame pairs in the tree. Formally, for the $n-$th frame embedding $f_{n,l} \in \mathbb{R}^{M \times D}$ on the l-th level, it needs to calculate attention with its parent $f_{\lfloor n/A \rfloor, l+1}$ and children $f_{(n-1)A+1:nA, l-1}$, where $0 \leq n \leq \lfloor N_V/(MA^l) \rfloor - 1$, and N_V/M equals the number of the original video frames retained in the previous frame pruning.

More importantly, in contrast to trivial image sequences with separate multiple images, long videos exhibit long-term temporal dependencies and latent causal relationships between frames, which are crucial for video understanding. Therefore, to allow each frame to capture the global semantics of the entire video, we additionally permit attention flows between E-hop neighbor frames at the same level. Formally, the frame embedding $f_{n,l}$ also needs to perform attention with its E-hop neighbors $f_{n-E:n+E,l}$ at the same level, where $0 \leq n \leq \lfloor N_V/(MA^l) \rfloor - 1$. While Fig. 2 shows the case of $E = 1$, a larger E is recommended in practice to enhance the ability of Hierarchical At-

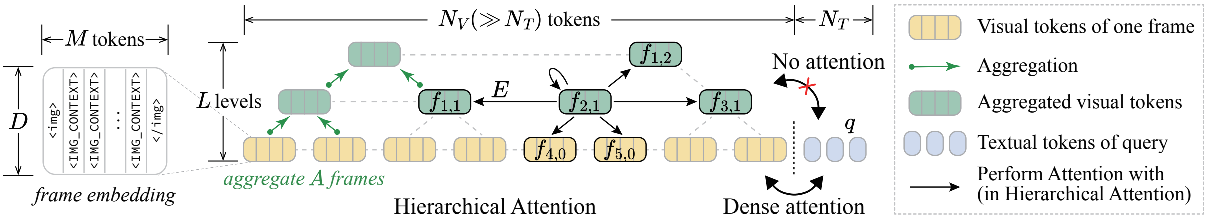

Figure 2: Overview of the Hierarchical Attention. First we hierarchically aggregate the visual tokens of original frames to obtain coarser-grained frames. Then, on the constructed tree (or trapezoid) structure, each frame only needs to perform attention operation with its parent, children, and the E-hop neighbors at the same level (the figure shows the condition where $E = 1$). By this way, the long-term temporal dependencies that are crucial in long videos can be captured more efficiently by the attention operation with the aggregated visual tokens of coarser frames.

tention to directly model short-term dependencies without increasing its theoretical complexity.

By combining these two strategies, the long-term temporal dependencies can be captured more efficiently by focusing on the aggregated coarser frames. This reduces the $\mathcal{O}(N_V^2)$ time and space complexity in the original attention to $\mathcal{O}(N_V)$, with theoretical proof provided below.

Lemma 1. *The introduced Hierarchical Attention has a time and space complexity of $\mathcal{O}(N_V)$ w.r.t. the visual sequence length N_V.*

Proof. For the $n-$th frame embedding $f_{n,l}$ on the l-th level, it only needs to perform attention operations with at most $(2E+1)+A+1$ frames, where M^2 calculations[1] are required between each two frames. Therefore, the amount of calculation C_l required by all the frames $f_{:,l}$ on the l-th level is:

$$C_l \leq \lfloor \frac{N_V}{MA^l} \rfloor (2E + A + 2)M^2.$$

The total computational cost C across all levels is:

$$C = \sum_{l=0}^{L-1} C_l \leq \sum_{l=0}^{L-1} \lfloor \frac{N_V}{MA^l} \rfloor (2E + A + 2)M^2$$
$$\leq N_V(2E + A + 2)M \sum_{l=0}^{L-1} \frac{1}{A^l} = \mathcal{O}(N_V)$$

Lemma 2. *Under appropriate parameter conditions, the Hierarchical Attention has an IPD of $\mathcal{O}(1)$ between any two frames, i.e., any frame can obtain the most distant information through an information propagation path of length $\mathcal{O}(1)$.*

Proof. Please refer to App. B for this proof.

In summary, as shown in lemma 1, the Hierarchical Attention on frame tokens reduces the first $\mathcal{O}(N_V^2)$ term in Formula 3 to $\mathcal{O}(N_V)$. This may still maintain an IPD of $\mathcal{O}(1)$ between frames, thereby theoretically keeping the model's capability for video understanding (lemma 2). As for the text query, we apply the quadratic attention between textual tokens and the original frame tokens (at the $l = 0$ level) to enhance the modeling of text-video correlations (Fig. 2). Notably, textual tokens do not directly attend to any aggregated visual tokens, as the latter are solely designed to facilitate efficient sparse attention between frames (Fig. 2).

3 Experiment

3.1 Experimental Setup

Attention Implementation. For the Hierarchical Attention, a simple approach is to flatten the aggregated frame embeddings $f_{:,l}(l \neq 0)$ into a sequence, and concatenate this with the original frame embeddings $f_{:,0}$, resulting in a sequence of length $\Sigma_{l=0}^{L-1} \lfloor N_V/(MA^l) \rfloor$. Then, all attention operations of the Hierarchical Attention can be performed on this extended sequence using a special attention mask. However, its complexity still remains $\mathcal{O}(N^2)$. To solve this issue, we *specifically implemented a CUDA kernel* using Triton (OpenAI, 2021), which actually achieves the complexity of $\mathcal{O}(N)$ (details in Sec. 3.3). Like sliding window attention, our Hierarchical Attention allows long-range token interaction through stacked multi-step attention operations. The hyperparameter values of α, β, E, A, and L are set as 30, 30, 2, 2, and 4 respectively.

Training. Our training is based on the pre-trained vision encoder and LLM (specifically InternViT-300M and InternLM2.5-7B (Cai et al., 2024) in our experiments). To make LLM better adapt to the new modules we introduced, we first uniformly

[1] The dense attention is still performed among the M visual tokens within each frame, because they do not exhibit partial order dependency and jointly represent one frame's semantics.

Table 1: Results comparison on long video understanding benchmarks. We mark the values ranking the first(**v**), second(v), and third(*v) in each column of the open-source MLLMs; and the first place (**v**) of the proprietary ones.

Models	Size	Ego-Schema	Movie-Chat-1k	Video-MME		Long-Video-Bench	LV-Bench	VN-Bench	MLVU	Event-Bench
				long	overall					
Proprietary MLLMs										
GPT-4V (OpenAI, 2023)	-	55.6	-	56.9	60.7	59.1	-	48.9	49.2	32.7
GPT-4o (OpenAI, 2024)	-	**72.2**	-	65.3	71.9	**66.7**	27.0	64.4	**64.6**	**53.3**
Gemini 1.5 Pro (Google, 2024)	-	63.2	-	**67.4**	**75.0**	64.0	**33.1**	**66.7**	-	43.2
Open-source MLLMs										
Video-LLaVA (Lin et al., 2023)	7B	38.4	-	36.2	39.9	39.1	-	12.4	47.3	5.9
Video-LLaMA (Cheng et al., 2024)	7B	51.7	51.7	42.1	47.9	-	-	4.5	35.5	6.7
Video-ChatGPT (Maaz et al., 2023)	7B	-	47.6	-	-	-	-	4.1	31.3	11.8
ShareGPT4Video (Chen et al., 2024a)	8B	-	-	35.0	39.9	39.7	-	-	46.4	-
MovieChat (Song et al., 2024)	7B	53.5	62.3	33.4	38.2	-	22.5	-	25.8	16.2
LLaVA-Next-Video (Zhang et al., 2024a)	34B	43.9	-	-	46.5	50.5	32.2	20.1	33.7	-
VideoChat2 (Li et al., 2024)	7B	55.8	-	33.2	39.5	39.3	-	12.4	47.9	29.4
PLLaVA (Xu et al., 2024)	34B	54.4	-	-	-	53.2	26.1	-	-	33.2
Kangaroo (Liu et al., 2024b)	8B	62.7	-	46.6	56.0	54.8	*39.4	-	61.0	-
MiniCPM-V-2.6 (Yao et al., 2024)	8B	46.5	82.9	51.8	60.9	54.9	25.9	22.0	48.5	57.3
LongLLaVA (Wang et al., 2024c)	7B	-	72.9	45.4	52.9	-	-	52.1	-	-
Video-XL (Shu et al., 2024)	7B	-	-	49.2	55.5	49.5	-	**61.6**	*64.9	-
TimeMarker (Chen et al., 2024b)	8B	-	-	46.4	57.3	56.3	**41.3**	-	63.9	-
LongVU (Shen et al., 2024)	7B	67.6	-	59.5	60.6	-	-	-	65.4	-
Qwen2-VL (Wang et al., 2024a)	7B	*66.7	*75.1	-	**63.3**	*55.6	35.8	33.9	48.5	*62.3
InternVL2 (Chen et al., 2024e)	40B	56.4	71.8	*52.6	*61.2	**59.3**	39.6	34.1	44.8	**67.6**
Sophia (ours)	8B	**79.2**	**86.8**	**59.6**	62.5	**59.3**	**41.3**	*38.0	**67.1**	63.8

sample frames and train the CNN of Hierarchical Attention from scratch using a constant learning rate of 3×10^{-4} with 8 GPUs and a global batch size of 128, while all other parameters remain frozen. Then, the aggregation CNN and LLM are jointly trained with a learning rate of 1×10^{-5}, using the cosine scheduler with a warmup rate of 0.03 on 16 GPUs and a global batch size of 128. After that, the Shot-adaptive Frame Pruning module is solely trained with a learning rate of 1×10^{-4} to allow the learning to select frames relevant to queries. The entire training is conducted using 192 CPUs and 16 GPUs (NVIDIA A100 80G), taking about four days. All the training is conducted using the SWIFT (Zhao et al., 2025) framework. The training data contains video data of various lengths, including ActivityNetQA (Yu et al., 2019), PerceptionTest (Pătrăucean et al., 2023), NeXT-QA (Xiao et al., 2021), Youcook2 (Zhou et al., 2017), MovieChat-1K-train (Song et al., 2024), and LLaVA-Video-178K (Zhang et al., 2024b). More details about the contributions of each training dataset are given in App. C.

Evaluation. To evaluate our model, we conduct extensive experiments on a wide range of benchmarks, including 7 recent long video benchmarks: EgoSchema (Mangalam et al., 2023), MovieChat-1K (Song et al., 2024), LongVideoBench (Wu et al., 2024), LVBench (Wang et al., 2024b), VNBench (Zhao et al., 2024), MLVU (Zhou et al., 2025), and Event-Bench (Du et al., 2024); and 1 comprehensive video benchmark: Video-MME (Fu et al., 2024) (without subtitles). The length of these videos can be up to hours long, which can well evaluate the model capability for long video understanding. Details of the baselines are in App. D.

3.2 Experimental Results

The performance comparison of Sophia and the SOTA baselines for long video understanding is shown in Tab. 1. Overall, Sophia achieves the best performance on 6 out of 8 benchmarks, demonstrating competitive capabilities with existing SOTA methods. Specifically, Sophia brings 17.2%, 4.7%, 5.3%, and 4.3% relative improvements on the EgoSchema, MovieChat-1k, LongVideoBench, and LVBench, respectively. Notably, on the Video-MME benchmark which includes videos of various lengths, Sophia ranks second on the overall metric with a marginally lower performance by only 0.8%. However, it surpasses all other baselines on the long video metric, highlighting its robust long video understanding ability. Although our model ranks third on VNBench, it is worth noting that the top two baselines on VNBench do not

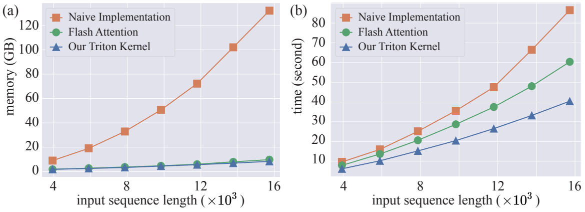

Figure 3: Memory and time consumption of different implementations of the Hierarchical Attention.

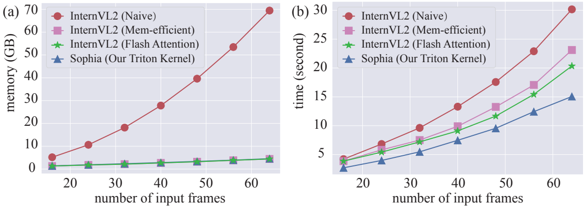

Figure 4: Memory and time consumption of our Triton kernel and different implementations of InternVL2-8B.

perform well on other benchmarks, which underscores the overall superiority and robustness of our model. In addition, Sophia even outperforms the best proprietary MLLMs on EgoSchema, LVBench, MLVU and Event-Bench, further illustrating its strong video comprehension ability. When compared to significantly larger baseline models such as LLaVA-Next-Video (34B), PLLaVA (34B), and InternVL2 (40B), Sophia with only 8B parameters consistently outperforms them across nearly all benchmarks, which highlights the effectiveness of Sophia for long video understanding. More analysis about the experimental results on Event-Bench and VNBench are given in App. E.

3.3 Memory and Time Efficiency

To illustrate the computational superiority of our CUDA kernel (Sec. 3.1) for Hierarchical Attention, we report the number of floating-point operations (FLOPs) of the attention modules of Sophia and the three best baselines in Tab. 2. Among these, Sophia consistently requires the fewest FLOPs across varying numbers of input frames. Additionally, as the number of frames increases, other baselines utilizing dense attention exhibit a rapid escalation in FLOPs, whereas Sophia maintains a steady and linear growth, which demonstrates the efficiency of our Triton kernel in computing resources. The total inference FLOPs of these models are in App. F.

To further empirically evaluate the Triton kernel we build, we conduct the following two experiments. First, for Sophia, we compare our kernel against two implementations of the simple approach mentioned in Sec. 3.1: naive implementation and Flash Attention (Dao et al., 2022). Specifically, we run the forward propagation of attention 100 times and record the memory and time costs. As in Fig. 3(a), our kernel greatly reduces memory usage compared to the naive one and slightly outperforms Flash Attention for longer sequences. As for time, Fig. 3(b) shows that our kernel is obviously faster than the simple approach, and brings up to a 49.4% speedup relative to Flash Attention.

Table 2: Comparisons on attention FLOPs of the best three baselines and Sophia for different frame numbers.

#Frame	LongVU	Qwen2-VL-7B	Intern-VL2-8B	Sophia (our kernel)
32	6.86T	1.82T	2.22T	**0.66T**
64	23.65T	5.61T	6.68T	**1.32T**
128	87.03T	19.06T	22.33T	**2.64T**

Moreover, we compare our Sophia with three attention implementations of the InternVL2-8B in a way similar to the previous comparison, including (1) a naive implementation; (2) Memory-efficient Attention (Gschwind et al., 2023), and (3) Flash Attention. As shown in Fig. 4(a), the memory usage of our Triton kernel is on par with both Memory-efficient Attention and Flash Attention, which is significantly lower than the naive PyTorch implementation. Regarding execution time, Fig. 4(b) shows that our kernel consistently outperforms all implementations of InternVL2, bringing speedups of 38.2% and 28.6% compared to Memory-efficient and Flash Attention, respectively.

3.4 Model Analysis

Ablation Study. We conduct ablation experiments on three model variants, including the Sophia without Frame Pruning (Sophia w/o Pruning), Sophia without the Hierarchical Attention (Sophia w/o HierAtt), and Sophia without both of the two modules (Sophia w/o both). As shown in Tab. 3, all the three variants perform worse than the full model, demonstrating the effectiveness of our proposed techniques for noise pruning and long-term dependency capturing in long video understanding. The FLOPs comparison of these variants is in App. G.

Table 3: The ablation study on two model variants.

Sophia	w/o both	w/o Pruning	w/o HierAtt	full
Video-MME overall	54.0	58.1	57.6	**62.5**

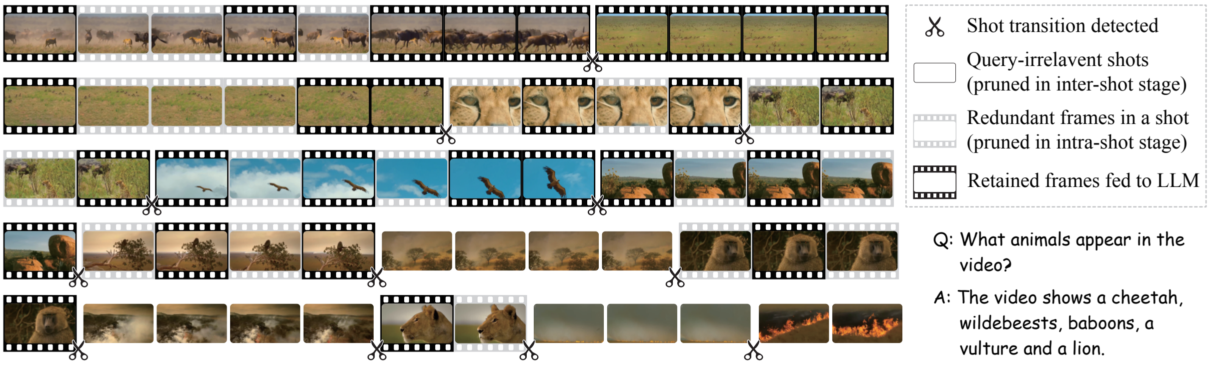
Shot transition detected
Query-irrelavent shots (pruned in inter-shot stage)
Redundant frames in a shot (pruned in intra-shot stage)
Retained frames fed to LLM
Q: What animals appear in the video?
A: The video shows a cheetah, wildebeests, baboons, a vulture and a lion.

Figure 5: The intermediate and final results of our two-stage Shot-adaptive Frame Pruning technique.

Hyperparameter Analysis. To explore the effect of the parameter E on the Hierarchical Attention, we evaluate the performance of Sophia across various E values on the Video-MME (Tab. 4). As E increases, the experiment results show an overall increasing trend. This is because a higher E allows each frame to perform attention operations with more neighbors at the same level, which strengthens Sophia's ability to directly model the dependencies between frames (as pointed out in Sec. 2.2). In addition, the performance gains from increasing E are relatively small, demonstrating that the Hierarchical Attention effectively models the long-term dependencies for long video understanding. Therefore, the default value of $E = 2$ in our experiment is basically sufficient, eliminating the need to increase E and incur additional computation costs. The details about more parameters are in App. H.

Table 4: Results of various values of hyperparameter E.

value of E	2 (default)	3	4	5
Video-MME overall	62.5	62.9	63.7	64.0

Needle in a Haystack (NIAH). To assess the retrieval ability of Sophia from long videos, we further conduct the NIAH (Zhao et al., 2024) experiments. The results and analysis are in App. I.

3.5 Case Study

Fig. 5 shows the results of our two-stage Shot-adaptive Frame Pruning technique. First, the shot detector in our Sophia accurately segments a long video into several shots. In the inter-shot pruning stage, Sophia can effectively identify and remove shots that are irrelevant to the query's core semantics (such as the four scenes of burning grassland that do not pertain to "animals"). Then, in the intra-shot stage, Sophia prunes redundant frames within each shot caused by continuous actions or scenes. This case clearly illustrates that Sophia can segment long videos based on actual shot boundaries rather than fixed-length, and prunes noisy shots and redundant frames at two granularities, thereby ensuring efficient understanding of long videos.

4 Related Work

Sparse and Hierarchy-inspired Attention. To efficiently handle long input sequences, many researchers studied diverse sparse or hierarchy-style attention mechanisms. Beltagy et al. (2020) propose dilated or global sliding window attention to reduce the computational complexity for long document tasks. Chalkidis et al. (2022) develop a hierarchical attention Transformer that uses segment-wise followed by cross-segment encoders for long document classification. Liu et al. (2022) propose a pyramidal attention to handle the long-range dependencies for efficient time series data modeling. Hatamizadeh et al. (2023) employ a window-based hierarchical attention to achieve high image throughput for computer vision. However, most of them are limited to modalities on pure text or a single image, and few explored these sparse attention strategies on videos. To our knowledge, we are the first to employ the Hierarchical Attention for the long video understanding task.

Long Video Understanding. To address the long context challenge in long videos, some studies (Song et al., 2024; He et al., 2024) initially employ memory banks to store historical video frames. Other works focus on reducing the number of tokens of each frame to limit sequence length while keeping the frame sampling rate. Jin et al. (2024) employ dynamic visual tokens by merging K-nearest neighbor tokens of each frame. Lee et al. (2024) explore various video token merging strategies for video classification. Wang et al. (2024c) adopt a hybrid of Mamba (Gu and Dao, 2024) and Transformer (Vaswani et al., 2017) as model architecture for multiple-image sequences. Shu et al. (2024) condense visual contexts into highly compact forms for hour-scale video understanding. Chen et al. (2024b) propose a video model for dialogue with emphasized temporal localization.

However, the methods above inevitably compromise the model's grasp of details in long videos. Another insightful idea recognizes that processing the whole long videos is impractical, and instead focuses on pruning the frames irrelevant to the query. Wang et al. (2024d) propose a framework that bootstraps MLLMs with advanced temporal grounding capabilities for video understanding. Shen et al. (2024) utilize cross-modal query for frame feature reduction, and perform spatial token reduction based on temporal dependencies for long videos. Ataallah et al. (2024) retrieve the instruction-relevant video clips with LLM-generated text descriptions of video clips for video understanding. Although these works may reduce redundancy in long videos, most of them cut videos into clips of equal length when dropping, ignoring that the events in videos are dynamic and uneven in time. To address this, we are the first to propose to divide videos naturally based on shots to identify the query-relevant frames more sharply.

5 Conclusion

In this paper, we are the first to propose a shot-adaptive frame pruning technique and a Hierarchical Attention mechanism, which first prunes redundant frames and then models long videos with linear complexity. Experimentally, our Sophia shows competitive capability on various benchmarks including hour-long videos with minimal time and memory costs, which suggests that Sophia reveals a direction for efficient long video understanding.

6 Limitations

Although our Sophia achieves competitive experimental results with relatively low time and memory costs on the long video understanding task for videos up to one or two hours in length, its performance on even longer videos has yet to be evaluated due to the video length limitations of existing datasets and benchmarks. In the future, by collecting longer video data and expanding the scale of the training dataset, we aspire to fully unlock the potential of our model in comprehending extended-length videos, thereby propelling the evolution of multimodal large language models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kirolos Ataallah, Xiaoqian Shen, Eslam Abdelrahman, Essam Sleiman, Mingchen Zhuge, Jian Ding, Deyao Zhu, Jürgen Schmidhuber, and Mohamed Elhoseiny. 2024. Goldfish: Vision-language understanding of arbitrarily long videos. *arXiv preprint arXiv:2407.12679*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. Internlm2 technical report.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. 2024a. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. 2024b. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. 2024c. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2024d. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2024e. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*.

Yifan Du, Kun Zhou, Yuqi Huo, Yifan Li, Wayne Xin Zhao, Haoyu Lu, Zijia Zhao, Bingning Wang, Weipeng Chen, and Ji-Rong Wen. 2024. Towards event-oriented long video understanding. *arXiv preprint arXiv:2406.14129*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Gemini Team Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Michael Gschwind, Driss Guessous, and Christian Puhrsch. 2023. Accelerated pytorch 2 transformers. https://pytorch.org/blog/accelerated-pytorch-2/.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces.

Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2023. Fastervit: Fast vision transformers with hierarchical attention. *arXiv preprint arXiv:2306.06189*.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. 2024. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. 2024. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710.

Kamradt. 2023. Llm test: Needle in a haystack. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. 2024. Video token merging for long-form video understanding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. 2024b. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. 2022. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *The Thirty-seven Annual Conference on Neural Information Processing Systems*.

OpenAI. 2021. Introducing triton: Open-source gpu programming for neural networks. https://openai.com/index/triton/.

OpenAI. 2023. Gpt-4v system card. https://openai.com/index/gpt-4v-system-card/.

OpenAI. 2024. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/.

Pinelopi Papalampidi, Skanda Koppula, Shreya Pathak, Justin Chiu, Joe Heyward, Viorica Patraucean, Jiajun Shen, Antoine Miech, Andrew Zisserman, and Aida Nematzdeh. 2024. A simple recipe for contrastively pre-training video-first encoders beyond 16 frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14386–14397.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. 2023. Perception test: A diagnostic benchmark for multimodal video models. *arXiv preprint arXiv:2305.13786*.

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

Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. 2024. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

Tomás Soucek and Jakub Lokoc. 2020. Transnet V2: an effective deep network architecture for fast shot transition detection. *CoRR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *The Thirty Annual Conference on Neural Information Processing Systems*.

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

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024b. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*.

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. 2024c. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*.

Yuxuan Wang, Yueqian Wang, Pengfei Wu, Jianxin Liang, Dongyan Zhao, Yang Liu, and Zilong Zheng. 2024d. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9972–9987.

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

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa:next phase of question-answering to explaining temporal actions. *arXiv preprint arXiv:2105.08276*.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. 2022. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. *arXiv preprint arXiv:1906.02467*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Y Zhang, B Li, H Liu, Y Lee, L Gui, D Fu, J Feng, Z Liu, and C Li. 2024a. Llava-next: A strong zero-shot video understanding model. https://llava-vl.github.io/blog/2024-04-30-llava-next-video.

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

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. 2025. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29733–29735.

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. 2024. Needle in a video haystack: A scalable synthetic evaluator for video mllms. *arXiv preprint arXiv:2406.09367*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025. Mlvu: Benchmarking multi-task long video understanding. *arXiv preprint arXiv:2406.04264*.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2017. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv:1703.09788*.

A License Statement

The scientific artifacts used in this work are all publicly accessible and this work only uses them for research purposes, thus not violating any of the artifacts' licenses. The new model released in this work is also licensed for research purposes only, prohibiting any other misuse.

B The Proof of the Lemma 2

Proof. Firstly, to ensure that all the frames can get global information through multiple stacked attention layers, the two frames at the two ends of the topmost level (the level with index $L - 1$) must be able to exchange information. More specifically, the two frames $f_{0,L-1}$ and $f_{\lfloor \frac{N_V}{MA^{L-1}} \rfloor - 1, L-1}$ must be able to receive the information from each other through multiple stacked attention layers. Assuming there are K stacked attention layers in the model architecture, since each attention layer expands the receptive field by E adjacent frames, the overall receptive field after K layers becomes EK. Therefore, to achieve this global perception ability, the appropriate values of E, A, L need to satisfy the following condition 4:

$$EK \geq \lfloor \frac{N_V}{MA^{L-1}} \rfloor - 1. \tag{4}$$

For any two original frames $f_{n,0}$ and $f_{n',0}$ at the 0-th level, we use $S(f_{n,0} \rightarrow f_{n',0})$ to denote the length of the shortest information propagation path between these two frames. Then, under the condition 4, the maximum length of their shortest information propagation path is that between the two ends:

$$\begin{aligned}
&\max_{n,n'} S(f_{n,0} \rightarrow f_{n',0}) \\
&= S(f_{0,0} \rightarrow f_{-1,0}) \\
&\leq S(f_{0,0} \rightarrow f_{0,L-1} \rightarrow f_{-1,L-1} \rightarrow f_{-1,0}) \\
&= L - 1 + \lceil \frac{1}{E} (\lfloor \frac{N_V}{MA^{L-1}} \rfloor - 1) \rceil + L - 1 \\
&\leq 2L - 2 + K \\
&= \mathcal{O}(1)
\end{aligned}$$

Here we use the negative index -1 to denote the index of the rightmost frame in a certain level, i.e., $\lfloor N_V / (MA^{l-1}) \rfloor - 1$.

C Analysis of the Contribution of Each Training Dataset

Here we analyze the contribution of each training dataset as follows:

- ActivityNetQA, PerceptionTest, NeXT-QA, LLaVA-Video-178K: These are classic video understanding datasets covering tasks such as video captioning, multiple-choice QA, and open-ended QA. They provide semantic knowledge related to various model capabilities, from basic scene comprehension to causal action reasoning. Their main contribution is establishing Sophia's fundamental video understanding ability.

- YouCook2: A dataset focused on video procedure segmentation, mainly featuring cooking process steps and textual descriptions. Training with this dataset enhances Sophia's temporal grounding ability, particularly in modeling temporal dependencies and causal relationships.

- MovieChat-1K-train: A long-video understanding dataset composed of 15 different movie genres, supporting tasks such as global video captioning, global-level QA, and frame-specific QA. Its main contribution is further strengthening Sophia's capability in extreme long-video understanding.

D Details of Baselines

First, we compare Sophia with the existing methods for video understanding. The details of these baseline models are as follows:

- Video-LLaVA (Lin et al., 2023): This work advances the foundational LLM towards a unified large vision-language model by unifying visual representation into the language feature space.

- Video-LLaMA (Cheng et al., 2024): A set of video large language models that incorporate a Spatial-Temporal Convolution connector and an Audio Branch to effectively capture the intricate spatial and temporal dynamics of video data.

- Video-ChatGPT (Maaz et al., 2023): A multimodal model that merges a video-adapted visual encoder with an LLM to address the underexplored field of video-based conversation.

- ShareGPT4Video (Chen et al., 2024a): A series of models that aims at facilitating the video understanding of large video-language models and the video generation of text-to-video models via dense and precise captions.

- MovieChat (Song et al., 2024): A long video understanding method that takes advantage of the memory model, with tokens in Transformers being employed as the carriers of memory in combination with memory mechanism.

- LLaVA-Next-Video (Zhang et al., 2024a): A video understanding model that improves upon LLaVa-NeXT by fine-tuning on a mix if video and image dataset thus increasing the model's performance on videos.

- VideoChat2 (Li et al., 2024): A video MLLM baseline developed by progressive multimodal training with diverse instruction-tuning data.

- PLLaVA (Xu et al., 2024): This work proposes a pooling strategy to smooth the feature distribution along the temporal dimension and thus reduce the dominant impacts from the extreme features for video QA and captioning tasks.

- Kangaroo (Liu et al., 2024b): This work develops a data curation system to build a large-scale dataset with high-quality annotations for vision-language pre-training and instruction tuning.

- MiniCPM-V-2.6 (Yao et al., 2024): An efficient MLLM deployable on end-side devices developed by integrating the common MLLM techniques in architecture, pretraining and alignment.

- LongLLaVA (Wang et al., 2024c): A long-context MLLM that adapts a hybrid of Mamba and Transformer architectures, approach data construction with both temporal and spatial dependencies among multiple images.

- Video-XL (Shu et al., 2024): An efficient video understanding model for hour-scale videos that condenses visual contexts into highly compact forms and can process at most 2048 frames.

- TimeMarker (Chen et al., 2024b): A versatile Video-LLM designed for high-quality dialogue based on video content that integrates Temporal Separator Tokens to enhance temporal awareness, emphasizing temporal localization.

- LongVU (Shen et al., 2024): This work propose a spatiotemporal adaptive compression mechanism that reduces the number of video tokens while preserving visual details of long videos by leveraging cross-modal query and inter-frame dependencies.

- Qwen2-VL-7B (Wang et al., 2024a): This work presents the Qwen2-VL series, an advanced upgrade of the previous Qwen-VL, that redefines the conventional predetermined-resolution approach in visual processing. It also adopts several techniques like Naive Dynamic Resolution and Multimodal Rotary Position Embedding.

- InternVL2-8B (Chen et al., 2024e): This work introduces a MLLM to bridge the capability gap between open-source and proprietary commercial models in multimodal understanding. It also explores a continuous learning strategy for the large vision foundation model.

Also, we compare Sophia with the proprietary MLLMs. The details of these baseline models are:

- GPT-4V (OpenAI, 2023): A multimodal model developed by OpenAI that enables users to instruct GPT-4 to analyze image inputs.

- GPT-4o (OpenAI, 2024): A multilingual, multimodal generative pre-trained transformer developed by OpenAI that can process and generate text, images and audio.

- Gemini 1.5 Pro (Google, 2024): A multimodal model developed by Google that introduces a breakthrough context window of up to two million tokens — the longest context window of large scale foundation model yet.

E The analysis about the performance on Event-Bench and VNBench

Among the eight long video understanding benchmarks, Sophia ranked second and third on two benchmarks, Event-Bench and VNBench. Here we analyze the reasons why Sophia performed slightly lower on these two benchmarks.

These two benchmarks focus specifically on temporal grounding and temporal reasoning tasks, with higher reasoning complexity than others. For example, Event-Bench often includes multiple distracting events that challenge the model's ability to understand causal relationships between events. However, there is currently a lack of public training data tailored to such complex temporal reasoning tasks, which explains why Sophia does not achieve the top rank on these two benchmarks.

To address this limitation, in the future, we plan these targeted improvements. First, following the latest practice in Qwen2.5-VL (Bai et al., 2025), we will align Sophia's positional encoding with absolute timestamps to enhance its temporal awareness. Also, we will collect and construct more challenging training data for complex temporal grounding and reasoning tasks, further improving the model's capability in complex reasoning. Notably, while two baselines outperform Sophia on Event-Bench and VNBench, Sophia still achieves SOTA on six other benchmarks, demonstrating its strong overall performance in long-video understanding.

F FLOPs Comparison between Sophia and Baselines

In the Tab. 2 from Sec. 3.3, we evaluate the FLOPs of the single-layer attention modules of our model and those of three top-performing baselines, and find that our Sophia (with the Hierarchical Attention implemented using our custom Triton kernel) shows the lowest FLOPs, which underscores the efficiency of the LLM in our model to handle the video frames. As a further supplement, we compare the total inference FLOPs for these models, with the results summarized in Tab. 5.

As presented in Tab. 5, taking 128 input frames as an example, our model demonstrates obviously lower FLOPs in both the vision encoder[2] and LLM parts compared to the top three baselines. Therefore, our approach not only achieves competitive accuracy across various benchmarks (Sec. 3.2) but also reduces computational costs relative to other baselines, demonstrating the effectiveness of our shot-adaptive Frame Pruning technique and Hierarchical Attention mechanisms in facilitating efficient long video understanding.

G FLOPs Comparison of Ablation Study

As a supplement to Tab. 3 from Sec. 3.4, we further compare the inference FLOPs of Sophia and its two variants. As shown in Tab. 6, taking the FLOPs for sampling 128 frames as an example, both ablated versions of Sophia (without Frame Pruning and without Hierarchical Attention) exceed the FLOPs of the full Sophia model. Specifically, the Sophia w/o Pruning exhibits a 71.8% increase in FLOPs compared to the full model. This rise is attributed to the absence of the pruning of noisy shots and redundant frames, which forces the LLM to process a larger number of unnecessary frames. Also, the Sophia w/o HierAtt increases FLOPs by 28.9% compared to the full Sophia. This is because it loses our Hierarchical Attention mechanism that ensures linear complexity, preventing the LLM from efficiently handling the multitude of frames in long videos. Therefore, the Frame Pruning and the Hierarchical Attention techniques not only enable the Sophia model to achieve superior performance in long video understanding tasks, but also significantly reduce the computational costs during inference.

Table 5: Comparison on the inference FLOPs of Sophia and best three baselines for 128 frames.

Models	Vision	LLM	Total FLOPs
LongVU	228.7T	2998.0T	3226.7T
Qwen2-VL-7B	119.7T	783.2T	902.9T
InternVL2-8B	85.9T	904.0T	989.9T
Sophia	**85.9T**	**219.9T**	**305.8T**

Table 6: The comparisons of the experimental results and FLOPs of the ablation study on two model variants.

Sophia	w/o Pruning	w/o HierAtt	full
Video-MME overall	58.1	57.6	**62.5**
FLOPs (128 frames)	525.2T	394.3T	**305.8T**

H Analysis on More Hyperparameters

As a supplement to Tab. 4 from Sec. 3.4, we further explore the effects of the noisy shot pruning ratio in the inter-shot stage and the redundant frame pruning ratio in the intra-shot stage, denoted as α and β, on the performance of long video understanding tasks. As illustrated in Tab. 7, we first reducing α from the default value of 30 to 10, resulting in a slight decline in model performance. This decrease occurs because a lower α allows more redundant, query-irrelevant shots to be retained during frame pruning, diminishing the model's ability to extract critical information from long videos. Conversely, increasing α from 30 to 50 leads to a significant performance drop, indicating that excessive pruning removes too many shots, causing the model to

[2] Through experimental measurements, our lightweight shot detector requires only about 0.19T FLOPs of computation for processing 128 frames. This is much smaller than the computational amounts listed in Tab. 5 and can therefore be considered negligible.

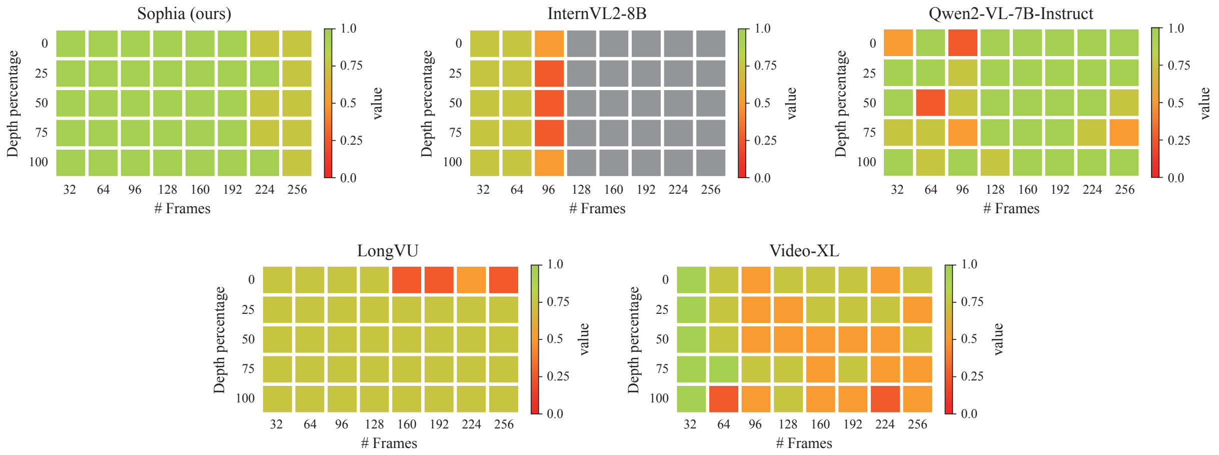

Figure 6: The results of the video needle-in-a-haystack experiment. The gray blocks denote cases where the input sequence length exceeds the LLM's context length limitations.

miss key frames essential for understanding long videos.

Regarding the analysis of β, as shown in Tab. 7, decreasing β value has little impact on model performance. This suggests that although more redundant frames are retained within each shot, these frames typically form a continuous sequence, and thereby minimally affect the semantic representation of videos. However, even though a lower β does not significantly alter the model's accuracy, it inevitably requires the LLM to process more input frames, resulting in increased memory usage and computation time. What's more, increasing β maintains model performance basically unchanged, demonstrating that our intra-shot frame pruning effectively eliminates redundant frames within shots, thereby extracting the critical information and enhancing the ability for long video understanding.

Table 7: Additional results of hyperparameter analysis on the Video-MME overall metric.

value of α value of β	30 30	10 30	50 30	30 10	30 50
Video-MME	62.5	61.7	58.8	62.6	62.1

I Needle in a Video Haystack

The needle-in-a-haystack (NIAH) (Kamradt, 2023; Hsieh et al., 2024) task is utilized to evaluate the ability of LLMs to retrieve information from long contexts. Following the practice of Shen et al. (2024) and Shu et al. (2024), we conduct the video needle-in-a-haystack experiment to compare the effectiveness of our Sophia and baselines in identifying the needle images from hour-scale long videos.

Specifically, this experiment involves inserting one irrelevant image (referred to as the "needle") into an hour-long video (referred to as the "haystack"). The task for the model is to locate the inserted needle image within the long video and answer related questions. The insertion points are set at 0% (beginning), 25%, 50% (midpoint), 75%, and 100% (end) of the video duration to assess the model's ability to retrieve needle images from various positions. We conduct our video NIAH experiment on four samples randomly selected from VNBench (Zhao et al., 2024), and the results are given in Fig. 6.

As shown in Fig. 6, we compare the performance of our Sophia model against the top baseline models on the video NIAH task. In addition to the top three baselines, we added a comparison with Video-XL, a baseline model that also claims to understand hour-long videos. In Fig. 6, the horizontal axis represents the total number of sampled frames in the video haystack, and the vertical axis indicates the insertion position of the needle image. The gray blocks of InternVL2-8B denote cases where the input sequence length of the LLM exceeds the model's context length limitations. Our findings reveal that, compared with the other baselines, our Sophia model can accurately identify the needle image within the video haystack, and effectively answer questions related to the needle image across various frame sampling rates and insertion positions. This demonstrates the superior capability of Sophia in processing and understanding long videos.

J The Discussion about Freezing the Shot Detector

As mentioned in Sec. 2.1, we chose to freeze the shot detector (TransNet) primarily for two reasons. First, TransNet is a well-balanced model in the scene transition detection field, achieving both efficiency and strong performance across various video scenarios. Second, in fields like computer vision, we note that placing a frozen module at the early part of a model is a common practice. For example, DETR (Carion et al., 2020) uses a frozen ResNet, and UniAD (You et al., 2022) adopts a frozen EfficientNet. Given these considerations, we opted to freeze TransNet to simplify the model training.

K The Discussion about Taking the Middle Frame as the Shot Representative

As mentioned in Sec. 2.1, to calculate the similarity, the embedding of the middle frame $f_{\lfloor (b_i+b_{i+1}-1)/2 \rfloor}$ of the shot $[r_{b_i}, r_{b_{i+1}-1}]$ is taken as the representative of the shot's semantic information. Here the reason we use the middle frame as the shot representative is two-fold. First, our primary motivation is that each shot in a long video typically conveys a relatively consistent semantic meaning (e.g., an ongoing action or a continuously displayed scene). Based on this assumption, our method does not explicitly account for semantic variations within the same shot. Additionally, in the extremely long video scenario we are targeting, where both the number of shots and frames is large, efficiency is crucial for constructing an effective long-video understanding method. To minimize computational overhead, we adopt a simple yet effective approach by selecting the middle frame as the representative. This helps streamline the model structure and improves the efficiency of semantic extraction.