VIDEOTREE: Adaptive Tree-based Video Representation for LLM Reasoning on Long Videos

Ziyang Wang* Shoubin Yu* Elias Stengel-Eskin*

Jaehong Yoon Feng Cheng Gedas Bertasius Mohit Bansal

Department of Computer Science UNC Chapel Hill https://videotree2024.github.io/

Abstract

Video-language understanding tasks have historically focused on short video clips, often struggling with the complexities of long-form video understanding. Recently, many long video-language understanding approaches have taken advantage of the reasoning capabilities of Large Language Models (LLMs) to perform long video question answering, transforming videos into densely sampled frame captions, and asking LLMs to respond to text queries over captions. However, the frames used for captioning are often redundant and contain irrelevant information, making dense sampling inefficient, and ignoring the fact that video question-answering requires varying levels of granularity, with some video segments being highly relevant to the question (and hence needing more fine-grained detail) while others being less relevant. Thus, these LLM-based approaches are prone to missing information and operate on large numbers of irrelevant captions, lowering both performance and efficiency. To address these shortcomings, we introduce VIDEOTREE, a queryadaptive and hierarchical framework for long-video understanding with LLMs. Specifically, VIDEOTREE dynamically extracts query-related information from the input video and builds a tree-based video representation for LLM reasoning. First, VIDEOTREE adaptively selects frames for captioning by clustering frames based on their visual features and scoring clusters based on their relevance to the query. We iterate this process until enough query-related keyframes are extracted. Second, it organizes visual clusters into a query-adaptive and hierarchical tree structure; the structure of the tree encodes varying levels of granularity, with higher (deeper) resolution on relevant segments. Finally, VIDEOTREE produces an answer to each question by traversing the tree's keyframes and passing their captions to an LLM answering model, which answers the query. Our experiments show that our trainingfree adaptive method improves both reasoning accuracy and efficiency compared to existing methods: VIDEOTREE achieves a 7.0%, 2.2%, and 2.7% improvement in accuracy over existing methods on the popular EgoSchema, NExT-QA, and IntentQA benchmarks, respectively, while reducing inference time by 40%.

1 Introduction

While recent developments in video-language understanding have led to major advances in answering questions about videos [79, 8, 7, 80, 4, 23], past work has largely focused on short video tasks, with videos typically ranging from 5 to 15 seconds [17, 70, 74]. Given the surge in accessible minutes-long video content and the importance of applications like long-form human behavior analysis

^{*}Equal contribution.



Figure 1: Overview of VIDEOTREE for LLM reasoning on long videos. VIDEOTREE helps to extract query-related information in long videos in a coarse-to-fine style. We first cluster videos via their visual embeddings, and then build a tree structure in a query-adaptive manner. Lastly, the LLM takes selected keyframe captions in the tree to conduct long video reasoning.

and movie analysis, developing models that can reason over and answer questions about long videos is increasingly crucial. However, most contemporary models [3, 79, 7, 76, 77] developed for short videos struggle with long-form understanding tasks. Recently, several approaches [83, 66] have emerged that leverage the long-sequence reasoning capabilities of Large Language Models (LLMs) to tackle the challenge of long-range modeling in long video-language understanding. Typically, these approaches leverage vision-language models (VLM) to caption densely sampled frames, thus representing the video in text format. This text representation is then subsequently fed into an LLM, which reasons over the video and responds to the provided query. Although this strategy has demonstrated great potentials on video understanding tasks, it still faces three major limitations that become particularly relevant when applied to *long-form video understanding* tasks:

1) Informational Overload: Long videos inherently contain high levels of information redundancy, and current long video understanding approaches [83, 10] lack a principled method to effectively address this challenge. A deluge of redundant information can overwhelm the LLM, leading to mistakes in video reasoning. As shown in Fig. 3, performance on video tasks actually decreases as the number of frames sampled increases.

2) Lack of Query Adaptability: Existing methods [83, 19] transform video inputs into textual descriptions without considering the query, resulting in irrelevant information being fed to the LLM. This is sub-optimal, and makes it harder for the LLM to accurately answer specific queries.

3) Inability to Capture the Coarse-to-Fine Video Structure: Existing approaches [83, 62] often simplify video content into a list of captions without any structure, failing to account for the hierarchical nature of video information. Especially in long videos, some video regions are information-dense – requiring fine-grained temporal understanding – while others are irrelevant to the query, or information-sparse. Because of this, dense sampling approaches not only suffer from the information overload problem (2) mentioned above, but also omit key information from the captions.

These limitations underscore the pressing need for a new long video understanding method. Specifically, the method should be *query-adaptive*, dynamically selecting parts of the video that are relevant to the question and ignoring those that are not, and *hierarchical*, i.e. able to capture different granularities of information, from coarse summaries to fine-grained actions. In other words, we would like a method that adaptively identifies key video segments and allocates more computational resources to these areas. To facilitate this, we represent the video as a tree structure. Specifically, we introduce **VIDEOTREE**, an adaptive tree-based method for long-video understanding. VIDEOTREE dynamically extracts query-related keyframes from the video input in a coarse-to-fine manner and organizes them within a tree structure, with child nodes representing more fine-grained information. Our sampling is *adaptive*, meaning that VIDEOTREE allocates more frames to relevant video regions and fewer samples to irrelevant ones. VIDEOTREE relies on three crucial steps: **adaptive breadth expansion** (Fig. 1a), **relevance-guided depth expansion** (Fig. 1b), and **LLM-based reasoning** (Fig. 1c). A more in-depth look at our method is given in Fig. 2; initially, we extract clusters of query-related keyframes to form the first level of tree nodes for the video representation (see Figure 2, Step 1). We adaptively increase

the number of keyframes through an iterative process of visual clustering, keyframe captioning, and relevance scoring until sufficient query-related information is gathered for LLM reasoning. After the initial clustering, for each first-level cluster, we extract its keyframe and determine the relevance of the cluster based on the keyframe's relevance to the query. We then expand deeper into the most relevant clusters to extract more detailed information (see Fig. 2, Step 2). For each cluster, we obtain a single keyframe and pass it to a VLM captioner. Finally, the keyframe captions from our tree representation are passed to an LLM, which reasons over them to answer the question (see Fig. 2, Step 3). This addresses the key problems with uniform dense sampling: VIDEOTREE is coarse-to-fine and conditioned on the query, making it not only more performant but also more computationally efficient, needing fewer frames to achieve stronger performance than densely-sampled frame baselines. Crucially, we demonstrate that the tree in VIDEOTREE provides a more relevant set of keyframes, and hierarchically represents coarse-to-fine information in an adaptive fashion. This significantly enhances the LLM's ability to reason effectively over long videos.

We demonstrate the effectiveness of VIDEOTREE by evaluating it on three standard long video question answering (LVQA) datasets: EgoSchema [40], which focuses on egocentric long-form video-language understanding; NExT-QA [73], a widely-used video question answering benchmark featuring videos that average 44 seconds in length; IntentQA [25], an LVQA dataset focused on reasoning about people's intent in long videos. Across these tasks, VIDEOTREE substantially improves accuracy compared to state-of-the-art LLM-based approaches, achieving an absolute accuracy improvement of 7.0% on EgoSchema benchmark, 2.2% on NExT-QA, and 2.7% on IntentQA. In our analysis, we show that our method results in the best of both worlds: better performance than strong baselines (e.g. [83]) and better efficiency. In summary, we present three major contributions:

- We propose *ViDeoTREE*, an adaptive tree-based representation for LLM reasoning over long-videos, which intelligently allocates a computational budget by dynamically extracting query-related keyframes from the video input in a coarse-to-fine manner.
- We show that using VIDEOTREE, we can extract frames that allow the LLM to understand and reason over the long videos better, resulting in substantial improvements over strong baselines across three standard long video question-answering datasets (EgoSchema, NExT-QA, and IntentQA).
- In our analysis, we find VIDEOTREE is not only more effective but also more efficient than uniform sampling baseline, achieving a 5.6% accuracy improvement using only 35% of the frame captions. Qualitatively, we find that VIDEOTREE selects query-relevant portions of the video to expand into, while glossing over irrelevant portions.

2 Related Work

Structural Video Representation. Video understanding [33, 59, 27, 15, 34, 6, 75, 31, 37, 68, 36, 57, 55, 71, 48, 58, 51, 45] has shown impressive advancement in both views of comprehension and efficiency. Recently, several video-language methods [1, 29, 16, 67, 82, 81, 46, 49, 78, 72] have further introduced a structured understanding of video frames to allow compact and efficient recognition of scene contexts. For example, HierVL [1] proposes a *bottom-up* hierarchical video-language embedding that capture video representations across short and long time periods. Specifically, HierVL performs contrastive learning between short clips and their captions, then aggregating representations for long video understanding tasks. VideoReCap [16] introduces a progressive video captioning approach that generates short clip-level captions and summarizes them into longer segments. These methods process long videos by progressively building high-level knowledge from local temporal information, i.e. in a bottom-up fashion that first captures all low-level details and then aggregates. This results in significant computational and time overhead. In contrast, VIDEOTREE employs a top-down approach with dynamic depth, enabling efficient and effective long video understanding by dynamically extracting query-related keyframes for LLM reasoning.

Video Understanding with LLMs. Inspired by the powerful reasoning capabilities of LLMs, recent works have explored using LLMs to address complex video-related tasks. Since LLMs primarily process text, various methods [42, 50, 32, 22, 69, 38, 84, 54, 5, 26, 18, 12, 14, 30, 65, 28] have been developed to efficiently train multimodal projectors to connect the visual encoder and LLMs or leverage caption-centric information. More specifically related to video reasoning, CREMA [80] integrates information from various modalities to address video reasoning problems. CREMA introduces a multimodal Q-former to project different modality features into the LLM embedding

space, enabling the LLM for the compositional understanding of videos to predict answers. While CREMA trains their Q-former, other past work [56, 20, 19, 11, 62, 53, 9, 60, 21] has investigated training-free combinations of captioners and LLMs for video understanding. Specifically, LLoVi [83] proposes a simple language-guided video understanding method. First, it extracts short-term video descriptions with a captioning model, and then an LLM summarizes these dense captions and responds to the given prompt. VideoAgent [62] introduces a multi-round frame search strategy using an LLM. This approach performs answer prediction and self-reflection based on the captions from previously collected frames and the input prompt, and explores the next frame until the LLM is satisfied. However, these methods struggle with high computational and monetary costs during long-video comprehension, as they require processing various types of input, generating dense captions for video frames or clips, or frequent LLM calls to perform self-evaluation and iterative frame selection. Instead, we propose to extract the key information from long videos by building an adaptive tree representation and sparsely selecting the keyframes for further LLM reasoning.

3 VIDEOTREE: Adaptive Tree-based Representation for Long Video-Language Understanding with LLMs

We present VIDEOTREE, an efficient and effective long video understanding framework that constructs a query-aware, query-adaptive hierarchical video representation for LLM reasoning over long videos. As illustrated in Fig. 2, the VIDEOTREE framework consists of three main steps: adaptive breadth expansion, relevance-guided depth expansion, and LLM video reasoning.

3.1 Adaptive Breadth Expansion

We build the first layer of the tree representation by finding key semantic information within the video. To extract key information, existing approaches [79, 61] select a fixed number of keyframes as the video representation. However, as mentioned in Sec. 1, this kind of uniform keyframe selection is sub-optimal for a general long video-language understanding framework, as it can miss information, includes redundant frames, and does not intelligently allocate the frame budget according to the video and query. To this end, we introduce an adaptive breadth expansion method that builds the first level of the tree representation. This method dynamically finds clusters of frames that are relevant to the query. Specifically, as shown in the left of Fig. 2 (Step 1), given the video and a query about it, we build the first level of the tree by iterating three operations: visual clustering, cluster captioning, and relevance scoring. These operations first group similar frames together, then assign captions to each cluster and determine how relevant each cluster is to the query. By iterating the process, VIDEOTREE *adaptively* allocates more computation to the most relevant regions of the video. In the following, we motivate and introduce each operation in detail.

Visual Clustering. To reduce frame redundancy, we first propose a visual clustering operation that groups similar frames together before extracting information from them. Specifically, given a video sequence $V = (F_1, F_2..., F_n)$, where F_i is the frame at the time step i and n is the length of the video, we extract visual features for each frame with the pre-trained visual encoder [52] E, such that $f_i = E(F_i)$, where $f_i \in \mathbb{R}^d$ is the visual representation extracted by the frame F_i . Note that this visual feature extraction is less expensive than extracting textual features like captions using a VLM since visual feature extraction does not involve autoregressive decoding, making it about 2 times faster. These features serve as a compact representation of each frame's visual content, capturing diverse semantics of each frame such as scenes and objects. Then we use K-Means clustering [39], to group frame features into k distinct clusters, which we denote as:

$$(C_1, C_2, \dots C_k), (c_1, c_2, \dots c_k) = \text{K-Means}((f_1, f_2, \dots, f_n), k)$$
 (1)

where, C_i is the *i*th cluster that groups multiple frames, c_i is the centroid vector for the *i*th cluster and k is the number of clusters, which can be thought of as the granularity of the tree's first level. This clustering process allows us to convert the video input into k clusters of similar frames.

Keyframe Captioning. To convert the cluster's visual features into textual information that can be processed by an LLM, we caption a single keyframe from each cluster. Specifically, for the cluster C_i , we find the keyframe F_i that is closest to the centroid vector c_i and consider it as the representative frame of the *i*th cluster. We then feed all extracted keyframes into the VLM-based captioner $Cap(\cdot)$ [85] and obtain a text caption $t_i = Cap(F_i)$ for each cluster.



[Question]: Instead of listing individual actions, summarize the process [QA Prompt]: Please provide the answer cused to handle the branches and maintain the trees in the video. with a single-letter (A, B, C, D, E). [Relev. Prompt]: rate your confidence level in this choice on a scale from 1 to 3, where 1 is lowest and 3 is highest.

Figure 2: A detailed view of VIDEOTREE. To construct the tree structure, we begin with *Adaptive Breadth Expansion* (Step 1), which dynamically extracts query-related key information by considering both video and question inputs. Then, starting from the highly relevant root nodes, we explore deeper into the tree branches with *Relevance-guided Depth Expansion* (Step 2), re-clustering at each level to capture finer visual cues. Finally, we gather the selected nodes (keyframes), caption them, and arrange them in temporal order for *LLM reasoning* (Step 3).

Relevance Scoring. After obtaining the keyframe captions t_i , we use the LLM to decide whether the extracted key information from each cluster is sufficient for the LLM to answer the given query. To this end, we feed all cluster captions $\{t_i \ \forall i \in [1, ..., k]\}$ from the last operation and the video query Q into the LLM and output a set of relevance scores $\{r_i \ \forall i \in [1, ..., k]\}$ for each cluster, where r_i is the relevance of the i_{th} cluster. To obtain each r_i , we prompt an LLM with the VLM captions and the query and ask it to assign each caption a relevance score of 1 to $max_relevance$, with 1 being least relevant. See Tab. 8 for all prompts.

Then, we set a maximum value for the number of clusters ($max_breadth$) and a requirement of the high-relevance cluster number (selected based on validation set) for adaptively extract the keyframe information within each cluster. If the number of high-relevance clusters is below the requirement, that indicates the information extracted from the current set of keyframes is insufficient for the LLM to answer the video query. In that case, we increase the number of clusters k and repeat the clustering, captioning, and relevance scoring operations. If the number of high-relevance clusters meets the requirements or the number of clusters reaches $max_breadth$, we append the extracted clusters with their keyframes to the first layer of the tree and continue to the next step (see more details in lines 21-26 in Algorithm 1).

3.2 Relevance-Guided Depth Expansion

Existing keyframe selection approaches [79, 61, 62] typically treat the selected frame as an unstructured list, neglecting potential internal structures within the video information. Specifically, these methods overlook that certain video regions are information-rich and require detailed sampling, whereas other areas, irrelevant to the query, may only need coarse or minimal sampling. The uniform or random sampling approach results in a long, redundant, and disorganized caption list that can confuse the LLM, ultimately limiting its ability to reason effectively and failing to provide a clear depiction of the video's structure. Thus, as shown in Step 2 of Fig. 2, we build a query-adaptive hierarchical video representation on top of the clusters from the first breadth expansion step. Specifically, we expand the depth of the tree according to relevance score of each cluster from the first step. The intuition is that for high-relevance clusters, the LLM requires more detailed, granular information,

while for low-relevance clusters, more information could actually lead to irrelevant details and could overwhelm the LLM, leading to incorrect reasoning.

The relevance of a top-level node (cluster) informs how many levels of more granular information we will extract from it, i.e. how many children/grandchildren the node will have. Since the relevance score r is one of $[1, 2, ..., max_relevance]$, we expand the tree based on the remaining relevance of the parent node. Thus, the value of $max_relevance$ is equal to the max depth max_depth of the tree-based video representation. Specifically, we do not expand nodes with low relevance ($r_i = 1$) further. For nodes with medium or high relevance (greater than 1), we re-cluster their constituent frames into w sub-clusters, where w denotes the branch-width of the tree, on the premise that these more relevant clusters should have more keyframes allocated to them. We then add those sub-clusters as children in the tree, giving them a relevance that is one less than their parent's. We recursively repeat this process until all nodes are leaf nodes with relevance 1 (see lines 2-12 in Algorithm 1). After the breadth and depth expansion on the tree, we obtain the adaptive hierarchical video representation for LLM reasoning over the long video.

3.3 LLM Video Reasoning

Finally, in order to use the LLM's ability on video reasoning, we need to present the LLM with a text-based video description. To this end, we traverse the nodes of the tree starting at the roots and expanding to the leaves, extracting keyframes from the tree's clusters at all levels and passing them into the VLM captioner to obtain keyframe captions. We then sort these keyframe captions in temporal order and concatenate them into a textual description of the video. Finally, we pass this description and the input query to the LLM and output the final answer (see line 38-40 in Algorithm 1) Our prompt is in Tab. 9.

4 Experimental Setup

Tasks & Datasets. We test our VIDEOTREE framework on three diverse long-form video questionanswering benchmarks: (1) **EgoSchema** [40], a long-range video question-answering benchmark consisting of 5K multiple choice question-answer pairs spanning 250 hours of video and covering a wide range of human activities. Our ablation studies are conducted on the official validation set of EgoSchema which contains 500 questions (referred to as the EgoSchema Subset). The videos are 180s long on average. (2) **NExT-QA** [73], a video question-answering benchmark for causal and temporal reasoning. It contains 5440 videos with an average length of 44s and approximately 52K questions. NExT-QA contains 3 different question types: Temporal (Tem.), Causal (Cau.), and Descriptive (Des.). (3) **IntentQA** [25] contains 4,303 videos and 16K multiple-choice question-answer pairs focused on reasoning about people's intent in the video. We perform a zero-shot evaluation on the test set containing 2K questions. The videos are more than 44s in average length.

Implementation Details. We adopt GPT-4¹ [43] as our LLM. Following VideoAgent [62], we leverage EVACLIP-8B [52] as our visual encoder and CogAgent [13] as the captioner for NExT-QA benchmark. Following LLoVi [83], we use LaViLa [85] as our captioner for the EgoSchema benchmark. Similar to [83], we use LLaVA1.6-7B [35] as our captioner for the IntentQA dataset. For clustering, we use kmeans_pytorch Library ². The best setting for *max_breadth*, *max_depth*, and *branch_width* on the EgoSchema validation set is 32, 3, and 4 and for NExT-QA and IntentQA, we set the hyper-parameter as 8, 3, and 2. We ablate the hyper-parameter choices in detail in Sec. A. We also provide additional implementation details in Appendix Sec. C, including detailed prompting information for VIDEOTREE.

Baselines. We compare VIDEOTREE with relevant baselines from previous work, including models based on video transformers [2, 44, 63, 64], open-source LLMs [79, 47], and those utilizing proprietary models, including the methods using GPT-3.5 [9], GPT4 [83, 11, 62], PaLM-2 [41] and GPT-4V [20].

Evaluation Metrics. We evaluate VIDEOTREE on all datasets under the multiple-choice QA setting. We utilize standard accuracy metrics for all experiments.

¹version 1106

²https://github.com/subhadarship/kmeans_pytorch

5 Experiments

5.1 Comparison with State-of-the-art Approaches

Table 1: Comparison with other methods on EgoSchema, NExT-QA, and IntentQA datasets. We compare our VIDEOTREE framework with three types of existing works, including video transformer models, open-source LLM-based models, and proprietary LLM-based models.

Model	LLM	EgoSchema		NExT-QA				IntentQA		
		Sub.	Full	Tem. Cau.		Des. Avg.		Full		
Video Transformer Models										
LongViViT [44]	-	56.8	33.3	-	-	-	-	-		
MC-ViT-L [2]	-	62.6	44.4	-	-	-	65.0	-		
InternVideo [63]	-	32.1	-	43.4	48.0	65.1	49.1	-		
InternVideo2 [64]	-	41.1	-	-	-	-	-	-		
Based on open-source LLMs										
SeViLA [79]	FlanT5-3B	25.7	22.7	61.3	61.5	75.6	63.6	60.9		
MVU [47]	Mistral-13B	60.3	37.6	55.4	48.1	64.1	55.2	-		
	В	ased or	ı proprie	etary LLN	As					
ProViQ [9]	GPT-3.5	57.1	-	-	-	-	64.6	-		
LLoVi [83]	GPT-4	57.6	50.3	61.0	69.5	75.6	67.7	64.0		
VideoAgent [62]	GPT-4	60.2	54.1	64.5	72.7	81.1	71.3	-		
VideoAgent [11]	GPT-4	62.8	-	-	-	-	-	-		
MoReVQA [41]	PaLM-2	-	51.7	64.6	70.2	-	69.2	-		
IG-VLM [20]	GPT-4V	59.8	-	63.6	69.8	74.7	68.6	64.2		
VIDEOTREE (Ours)	GPT-4	66.2	61.1	67.0	75.2	81.3	73.5	66.9		

Tab. 1 shows a comparison of the existing works and VIDEOTREE on three diverse video questionanswering benchmarks, including EgoSchema, NExT-QA, and IntentQA.

EgoSchema. On the long-form video question-answering dataset EgoSchema [40], our VIDEOTREE framework substantially outperforms the existing GPT-4-based approaches, including LLoVi [83] and VideoAgent [11, 62]. Specifically, we outperform the state-of-the-art VideoAgent methods [11, 62] with 3.4% and 7.0% improvements on the subset and full test set, verifying the effectiveness of the proposed adaptive tree-based representation on long video understanding. Moreover, VIDEOTREE outperforms the GPT-4V-based method IG-VLM [20], and surpasses strong multimodal LLM without using an expensive end-to-end multimodal model.

NExT-QA. On the NExT-QA benchmark, VIDEOTREE achieves 73.5% zero-shot accuracy on the validation set, outperforming existing state-of-the-art method VideoAgent [62] by 2.2% on average accuracy using the same captioner and LLM. Furthermore, VIDEOTREE again outperforms the GPT-4V-based method IG-VLM [20]. Notably, NExT-QA contains various video queries, including temporal modeling (Tem.), causal reasoning (Cas.), and descriptive questions (Des.). We show that the VIDEOTREE framework surpasses the existing approaches [83, 62] on all query types, indicating VIDEOTREE improves the LLM reasoning ability on long videos under different scenarios.

IntentQA. On IntentQA, our VIDEOTREE framework achieves 66.9% zero-shot accuracy on the test set, surpassing the existing approaches with 2.7% improvements. This result shows that VIDEOTREE improves performance in answering questions about intent, which is challenging since intent understanding [25] requires the model to understand the various video contexts, including the immediate communicative context, the shared experience, and the commonsense.

5.2 Analysis

Below, we provide a detailed analysis of our VIDEOTREE framework. All quantitative analyses are conducted on the validation subset of the EgoSchema dataset. First, we analyze the tradeoff between efficiency and effectiveness, comparing VIDEOTREE to the LLoVi baseline [83]. Here, we

show that our method has better efficiency *and* performance across all settings. We then verify the effectiveness of the query-adaptive hierarchical video representation by comparing against different alternative representations. Finally, we visualize the output trees from VIDEOTREE and show the clusters VIDEOTREE chooses to expand, qualitatively supporting its quantitative gains. We also provide an extensive ablation study (including hyper-parameter analysis and the design choices of VLM/LLM) in Appendix Sec. A.

5.2.1 Efficiency-Effectiveness Analysis

In Fig. 3, we plot the Pareto curve between efficiency and performance, i.e. we analyze the relationship between the efficiency (caption numbers) and effectiveness (EgoSchema performance) of the VIDEOTREE framework and compare with the baseline approach [83]. Specifically, using the same captioner and LLM, we compare VIDEOTREE, which features an adaptive tree-based video representation, against a uniform baseline. Specifically, we use the uniformly-sampled frame caption list from the LLoVi baseline [83]. Firstly, across all frame settings, VIDEOTREE outperforms LLoVi by 2.5% on average. This indicates that our framework substantially improves the LLM reasoning ability over long videos under different frame budgets. Furthermore, we also see that VIDEOTREE achieves comparable performance with the best baseline (32 frames) using only 15.6 frames on average, indicating the efficiency of the proposed framework. In other words, VIDEOTREE can do more with less, needing only half as many frames to achieve comparable performance.



Figure 3: Analysis of the efficiency and effectiveness relationship for VIDEOTREE. We compare our VIDEOTREE framework with LLoVi [83] under different frame number settings.

Table 2: Ablation for the effectiveness of the adaptive tree-based representation of the VIDEOTREE framework. We compare our Adaptive Tree representation (Ada-Tree) with the uniform sampling baseline (Baseline-64 and Baseline-180) and the static tree representation (Static Tree) which builds the tree representation with a fixed tree breadth and depth.

Representation	ES Acc↑	#Frame↓
Baseline-64	62.4	64
Baseline-180	60.6	90
Static Tree	63.8	112.8
Ada-Tree (ours)	66.2	63.2

5.2.2 Effectiveness of the Adaptive Tree Representation

In Tab. 2, we verify the effectiveness of the proposed adaptive tree-based representation on a long video understanding task, comparing with the uniform sampling baseline [83] and other tree representations. First, the results show that both tree representations (adaptive tree and static tree) outperform the uniform sampling baseline, indicating the importance of the hierarchical nature of the video representation. Furthermore, we compare our adaptive tree representation with a static tree variant, which is a tree representation obtained using the same visual clustering process as our adaptive tree-based representation but without any query adaptation. Instead of dynamically extracting query-related information using a relevance score, the static tree builds always use the maximum value of the width and depth for all branches, resulting in a full tree. This representation is hierarchical, but not adaptive. The results show that both tree representations (static/adaptive) outperform the uniformly sampled caption list, which confirms the importance of having a structural representation for long video understanding using LLM. We show the adaptive tree representation we use in VIDEOTREE outperforms the static tree in both efficiency (63.2 frames vs 112.8 frames) and effectiveness (66.2% vs 63.8%), highlighting the importance of having an efficient and query-related representation for LLM reasoning over long videos.

5.2.3 Qualitative Analysis



Figure 4: Qualitative examples of VIDEOTREE keyframes and captions selection. Red options are answered wrongly with uniformly sampled 32 frames. Green options are answered correctly with VIDEOTREE. Best viewed in color. We include additional visualization in Appendix Sec. D.

In Figure 4, we visualize qualitative results from VIDEOTREE. Specifically, we show the keyframes and their captions extracted by our adaptive tree representation given a video query. This example is drawn from EgoSchema, and shows the query format, which consists of a query and multiple-choice answers. With the proposed VIDEOTREE strategy, we can split a complex multi-scene video (*e.g.cleaning house across rooms*) into several key scenes via visual clustering and determine the most query-relevant scene via the relevance score. We then can obtain more fine-grained visual cues by descending into each relevant cluster (Levels 2 and 3 in Figure 4). For example "*C opens a washing machine*" is deemed highly relevant to the question, which asks about the sequence of events. At the same time, frames like "*C moves around*" are deemed irrelevant to the query and not expanded. In the end, VIDEOTREE shows a dynamic ability to select relevant segments and can answer the given question correctly with only 50% of the baseline's 32 input captions. The baseline (fixed uniformly sampling) fails to correctly answer the question, sampling a large number of redundant and irrelevant frames. We also provide additional qualitative results in Appendix Sec. D.

6 Conclusion

After noting three key problems with dense sampling, we proposed VIDEOTREE, an adaptive and hierarchical strategy for sampling frames, captioning them, and reasoning with an LLM to answer questions about long videos. VIDEOTREE resulted in substantial performance increases on three popular datasets (EgoSchema, NExT-QA, and IntentQA), while also improving efficiency by captioning fewer frames than uniform sampling. We analyzed the role of the adaptive cluster selection we implement in VIDEOTREE, finding that it is crucial to strong performance. In our qualitative analysis, we showed that given a complex multi-scene video and its query, our VIDEOTREE framework is capable of extracting key scenes and zooming into more detailed information that is highly related to the query.

Limitations and Broader Impact

Limitations. Like all LLM-based video-reasoning systems (including dense sampling) our method is limited by the ability of the captioner to accurately capture the contents of sampled frames. However, our method's modular nature means that as captioners improve, we can easily include them into the VIDEOTREE framework; similarly, we can use increasingly strong LLMs as the reasoning backbone of VIDEOTREE. While VIDEOTREE is training-free, it includes a small number of hyperparameters. In Sec. A, we ablate these hyperparameters, showing that VIDEOTREE outperforms the uniform-sampling baseline regardless of the choice of max depth and branch width. Thus, while better hyperparameters can benefit the method, even with suboptimal settings VIDEOTREE outperforms the uniform baseline.

Broader Impact. Our results indicate that we can have the best of both worlds: improved accuracy *and* improved efficiency. Given the importance of long video reasoning tasks, improving accuracy has obvious broader implications for building more usable video reasoning systems, which could contribute to a wide variety of positive applications. Efficiency improvements also contribute to the applicability of long video systems, as reducing latency and computational cost can speed up adoption. Furthermore, since both VLM captioners and LLM reasoners generally improve with increased scale, reducing the number of calls to them will become increasingly important; we expect the efficiency benefits coming from our method to play an even larger role in the future. Our work does not have any particularly relevant potential for negative applications or misuse beyond the standard caveats that apply to all machine learning systems.

Acknowledgements

We thank Ce Zhang, David Wan, and Jialu Li for their helpful discussions. This work was supported by DARPA ECOLE Program No. HR00112390060, NSF-AI Engage Institute DRL-2112635, DARPA Machine Commonsense (MCS) Grant N66001-19-2-4031, ARO Award W911NF2110220, ONR Grant N00014-23-1-2356, Sony Faculty Innovation award, Laboratory for Analytic Sciences via NC State University, and Accelerate Foundation Models Research program. The views contained in this article are those of the authors and not of the funding agency.

References

- K. Ashutosh, R. Girdhar, L. Torresani, and K. Grauman. Hiervl: Learning hierarchical videolanguage embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23066–23078, 2023.
- [2] I. Balažević, Y. Shi, P. Papalampidi, R. Chaabouni, S. Koppula, and O. J. Hénaff. Memory consolidation enables long-context video understanding. *arXiv preprint arXiv:2402.05861*, 2024.
- [3] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.
- [4] S. Buch, C. Eyzaguirre, A. Gaidon, J. Wu, L. Fei-Fei, and J. C. Niebles. Revisiting the "Video" in Video-Language Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [5] J. Chen, D. Zhu, K. Haydarov, X. Li, and M. Elhoseiny. Video ChatCaptioner: Towards enriched spatiotemporal descriptions, 2023.
- [6] F. Cheng and G. Bertasius. TALLFormer: Temporal action localization with a long-memory transformer, 2022.
- [7] F. Cheng, X. Wang, J. Lei, D. Crandall, M. Bansal, and G. Bertasius. VindLU: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10739–10750, June 2023.
- [8] F. Cheng, Z. Wang, Y.-L. Sung, Y.-B. Lin, M. Bansal, and G. Bertasius. DAM: Dynamic adapter merging for continual video QA learning, 2024.

- [9] R. Choudhury, K. Niinuma, K. M. Kitani, and L. A. Jeni. Zero-shot video question answering with procedural programs. arXiv preprint arXiv:2312.00937, 2023.
- [10] J. Chung and Y. Yu. Long Story Short: a summarize-then-search method for long video question answering, 2023.
- [11] Y. Fan, X. Ma, R. Wu, Y. Du, J. Li, Z. Gao, and Q. Li. Videoagent: A memory-augmented multimodal agent for video understanding. arXiv preprint arXiv:2403.11481, 2024.
- [12] B. He, H. Li, Y. K. Jang, M. Jia, X. Cao, A. Shah, A. Shrivastava, and S.-N. Lim. Ma-Imm: Memory-augmented large multimodal model for long-term video understanding, 2024.
- [13] W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Zhang, J. Li, B. Xu, Y. Dong, M. Ding, and J. Tang. CogAgent: A visual language model for GUI agents, 2023.
- [14] B. Huang, X. Wang, H. Chen, Z. Song, and W. Zhu. VTimeLLM: Empower LLM to grasp video moments, 2023.
- [15] S. Hwang, J. Yoon, Y. Lee, and S. J. Hwang. Everest: Efficient masked video autoencoder by removing redundant spatiotemporal tokens. In *International Conference on Machine Learning*, 2024.
- [16] M. M. Islam, N. Ho, X. Yang, T. Nagarajan, L. Torresani, and G. Bertasius. Video ReCap: Recursive captioning of hour-long videos. arXiv preprint arXiv:2402.13250, 2024.
- [17] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017.
- [18] P. Jin, R. Takanobu, W. Zhang, X. Cao, and L. Yuan. Chat-UniVi: Unified visual representation empowers large language models with image and video understanding, 2024.
- [19] K. Kahatapitiya, K. Ranasinghe, J. Park, and M. S. Ryoo. Language repository for long video understanding. arXiv preprint arXiv:2403.14622, 2024.
- [20] W. Kim, C. Choi, W. Lee, and W. Rhee. An image grid can be worth a video: Zero-shot video question answering using a vlm. arXiv preprint arXiv:2403.18406, 2024.
- [21] D. Ko, J. S. Lee, W. Kang, B. Roh, and H. J. Kim. Large language models are temporal and causal reasoners for video question answering, 2023.
- [22] B. Korbar, Y. Xian, A. Tonioni, A. Zisserman, and F. Tombari. Text-conditioned resampler for long form video understanding, 2024.
- [23] J. Lei, T. L. Berg, and M. Bansal. Revealing single frame bias for video-and-language learning, 2022.
- [24] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.
- [25] J. Li, P. Wei, W. Han, and L. Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings* of the IEEE/CVF International Conference on Computer Vision, pages 11963–11974, 2023.
- [26] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. VideoChat: Chat-centric video understanding, 2024.
- [27] K. Li, X. Li, Y. Wang, Y. He, Y. Wang, L. Wang, and Y. Qiao. Videomamba: State space model for efficient video understanding. arXiv preprint arXiv:2403.06977, 2024.
- [28] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo, L. Wang, and Y. Qiao. MVBench: A comprehensive multi-modal video understanding benchmark, 2024.
- [29] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu. HERO: Hierarchical encoder for video+language omni-representation pre-training, 2020.

- [30] Y. Li, X. Chen, B. Hu, and M. Zhang. LLMs meet long video: Advancing long video comprehension with an interactive visual adapter in LLMs, 2024.
- [31] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan. Video-Ilava: Learning united visual representation by alignment before projection, 2023.
- [32] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan. Video-Ilava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122, 2023.
- [33] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7083–7093, 2019.
- [34] K. Lin, F. Ahmed, L. Li, C.-C. Lin, E. Azarnasab, Z. Yang, J. Wang, L. Liang, Z. Liu, Y. Lu, C. Liu, and L. Wang. MM-VID: Advancing video understanding with GPT-4V(ision), 2023.
- [35] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/ 2024-01-30-llava-next/.
- [36] C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, and Z. Tu. Macaw-Ilm: Multi-modal language modeling with image, audio, video, and text integration, 2023.
- [37] F. Ma, X. Jin, H. Wang, Y. Xian, J. Feng, and Y. Yang. Vista-LLaMA: Reliable video narrator via equal distance to visual tokens, 2023.
- [38] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024.
- [39] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [40] K. Mangalam, R. Akshulakov, and J. Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36, 2024.
- [41] J. Min, S. Buch, A. Nagrani, M. Cho, and C. Schmid. MoReVQA: Exploring modular reasoning models for video question answering. arXiv preprint arXiv:2404.06511, 2024.
- [42] S. Munasinghe, R. Thushara, M. Maaz, H. A. Rasheed, S. Khan, M. Shah, and F. Khan. Pg-video-llava: Pixel grounding large video-language models. arXiv preprint arXiv:2311.13435, 2023.
- [43] OpenAI. GPT-4 technical report, 2023.
- [44] P. Papalampidi, S. Koppula, S. Pathak, J. Chiu, J. Heyward, V. Patraucean, J. Shen, A. Miech, A. Zisserman, and A. Nematzdeh. A simple recipe for contrastively pre-training video-first encoders beyond 16 frames, 2023.
- [45] L. Qian, J. Li, Y. Wu, Y. Ye, H. Fei, T.-S. Chua, Y. Zhuang, and S. Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning, 2024.
- [46] Z. Qing, S. Zhang, Z. Huang, Y. Xu, X. Wang, M. Tang, C. Gao, R. Jin, and N. Sang. Learning from untrimmed videos: Self-supervised video representation learning with hierarchical consistency, 2022.
- [47] K. Ranasinghe, X. Li, K. Kahatapitiya, and M. S. Ryoo. Understanding long videos in one multimodal language model pass, 2024.
- [48] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding, 2024.

- [49] K. Sanders, N. Weir, and B. V. Durme. TV-TREES: Multimodal entailment trees for neurosymbolic video reasoning, 2024.
- [50] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, X. Guo, T. Ye, Y. Lu, J.-N. Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint* arXiv:2307.16449, 2023.
- [51] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang, Y. Lu, J.-N. Hwang, and G. Wang. Moviechat: From dense token to sparse memory for long video understanding, 2024.
- [52] Q. Sun, J. Wang, Q. Yu, Y. Cui, F. Zhang, X. Zhang, and X. Wang. EVA-CLIP-18B: Scaling clip to 18 billion parameters. arXiv preprint arXiv:2402.04252, 2024.
- [53] D. Surís, S. Menon, and C. Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023.
- [54] R. Tan, X. Sun, P. Hu, J. hsien Wang, H. Deilamsalehy, B. A. Plummer, B. Russell, and K. Saenko. Koala: Key frame-conditioned long video-LLM, 2024.
- [55] A. Wang, B. Wu, S. Chen, Z. Chen, H. Guan, W.-N. Lee, E. L. Li, J. B. Tenenbaum, and C. Gan. Sok-bench: A situated video reasoning benchmark with aligned open-world knowledge. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [56] J. Wang, G. Bertasius, D. Tran, and L. Torresani. Long-short temporal contrastive learning of video transformers, 2022.
- [57] J. Wang, D. Chen, C. Luo, X. Dai, L. Yuan, Z. Wu, and Y.-G. Jiang. ChatVideo: A tracklet-centric multimodal and versatile video understanding system, 2023.
- [58] J. Wang, D. Chen, C. Luo, B. He, L. Yuan, Z. Wu, and Y.-G. Jiang. OmniVid: A generative framework for universal video understanding, 2024.
- [59] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, L. Yuan, and Y.-G. Jiang. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6312–6322, 2023.
- [60] S. Wang, Q. Zhao, M. Q. Do, N. Agarwal, K. Lee, and C. Sun. Vamos: Versatile action models for video understanding, 2023.
- [61] X. Wang, J. Liang, C.-K. Wang, K. Deng, Y. Lou, M. Lin, and S. Yang. Vila: Efficient video-language alignment for video question answering, 2024.
- [62] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. arXiv preprint arXiv:2403.10517, 2024.
- [63] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191, 2022.
- [64] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv* preprint arXiv:2403.15377, 2024.
- [65] Y. Wang, Y. Wang, P. Wu, J. Liang, D. Zhao, and Z. Zheng. LSTP: Language-guided spatial-temporal prompt learning for long-form video-text understanding, 2024.
- [66] Y. Wang, Y. Yang, and M. Ren. LifelongMemory: Leveraging LLMs for answering queries in long-form egocentric videos, 2024.
- [67] Z. Wang, Y.-L. Sung, F. Cheng, G. Bertasius, and M. Bansal. Unified coarse-to-fine alignment for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2816–2827, October 2023.

- [68] Z. Wang, L. Wang, Z. Zhao, M. Wu, C. Lyu, H. Li, D. Cai, L. Zhou, S. Shi, and Z. Tu. Gpt4Video: A unified multimodal large language model for Instruction-followed understanding and safety-aware generation, 2023.
- [69] Y. Weng, M. Han, H. He, X. Chang, and B. Zhuang. Longvlm: Efficient long video understanding via large language models, 2024.
- [70] B. Wu, S. Yu, Z. Chen, J. B. Tenenbaum, and C. Gan. Star: A benchmark for situated reasoning in real-world videos. arXiv preprint arXiv:2405.09711, 2024.
- [71] C.-Y. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer. MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition, 2022.
- [72] F. Xiao, K. Kundu, J. Tighe, and D. Modolo. Hierarchical self-supervised representation learning for movie understanding, 2022.
- [73] J. Xiao, X. Shang, A. Yao, and T.-S. Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision* and pattern recognition, pages 9777–9786, 2021.
- [74] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.
- [75] J. Xu, C. Lan, W. Xie, X. Chen, and Y. Lu. Retrieval-based video language model for efficient long video question answering. arXiv preprint arXiv:2312.04931, 2023.
- [76] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022.
- [77] A. Yang, A. Nagrani, P. H. Seo, A. Miech, J. Pont-Tuset, I. Laptev, J. Sivic, and C. Schmid. Vid2Seq: Large-scale pretraining of a visual language model for dense video captioning, 2023.
- [78] Z. Yang, G. Chen, X. Li, W. Wang, and Y. Yang. DoraemonGPT: Toward understanding dynamic scenes with large language models (exemplified as a video agent), 2024.
- [79] S. Yu, J. Cho, P. Yadav, and M. Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36, 2024.
- [80] S. Yu, J. Yoon, and M. Bansal. CREMA: Multimodal compositional video reasoning via efficient modular adaptation and fusion. arXiv preprint arXiv:2402.05889, 2024.
- [81] A. Zala, J. Cho, S. Kottur, X. Chen, B. Oguz, Y. Mehdad, and M. Bansal. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23056–23065, June 2023.
- [82] B. Zhang, H. Hu, and F. Sha. Cross-modal and hierarchical modeling of video and text. In Proceedings of the european conference on computer vision (ECCV), pages 374–390, 2018.
- [83] C. Zhang, T. Lu, M. M. Islam, Z. Wang, S. Yu, M. Bansal, and G. Bertasius. A simple LLM framework for long-range video question-answering. arXiv preprint arXiv:2312.17235, 2023.
- [84] H. Zhang, X. Li, and L. Bing. Video-Ilama: An instruction-tuned audio-visual language model for video understanding, 2023.
- [85] Y. Zhao, I. Misra, P. Krähenbühl, and R. Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023.

Appendix

In this Appendix, we present the following:

- Additional ablation study for our VIDEOTREE framework (Sec. A).
- The detailed algorithm for VIDEOTREE (Sec. B).
- Additional implementation details (Sec. C).
- Additional qualitative analysis (Sec. D).
- License information (Sec. E).

A Additional Ablation Study

In this section, we report additional ablation studies for our VIDEOTREE framework. First, we ablate the effect of the different tree settings for VIDEOTREE. Then, we analyze the effect of different VLM/LLM designs for VIDEOTREE.

Hyperparameter Analysis. In Tab. 3, we study the effect of the max depth of the tree-based representation for the VIDEOTREE. The results show that as depth increases, the performance increases, showcasing the importance of depth expansion. The max performance is reached when the max depth is set to 3, VIDEOTREE obtains the best performance. However, having a larger depth hurts VIDEOTREE's performance; note that all settings still outperform uniform sampling.

Table 3: The effect of different settings for max depth of VIDEOTREE. We show that when the max depth is set to 3, VIDEOTREE obtains the best performance on the subset of the EgoSchema dataset. We also show that decreasing the max depth can make the model more efficient while retaining strong performance, outperforming all existing approaches. Table 4: The effect of different settings for branch width of VIDEOTREE. The results show that when the branch width is set to 4, VIDEOTREE achieves the best performance on the EgoSchema subset. Reducing the branch width can make the model more efficient while retaining performance, outperforming all existing approaches.

Max Depth	ES Acc↑	#Frame↓	Branch Width	ES Acc↑	#Frame↓
1	63.4	32	2	64.4	43.5
2	64.6	46.5	3	65.0	54.6
3	66.2	63.2	4	66.2	63.2
4	62.6	89.1	5	64.2	72.5
Uniform Baseline	61.2	180	Uniform Baseline	61.2	180

Next, in Tab. 4, we study the effect of the branch width of the tree-based representation for the VIDEOTREE. The best performance is obtained when the branch width is set to 4. As with depth, excessive branch width reduces the VIDEOTREE performance due to the information overwhelming to the LLM; however, even with the worst branch width setting, VIDEOTREE still outperforms the baseline.

In Tab. 5, we study the effect of the max breadth of the adaptive tree-based representation. The results indicate that even with a smaller max tree breadth, VIDEOTREE achieves good performance while using much fewer frames. Increasing the breadth generally increases performance, with the best performance when the max breadth is set to 32. However, having an excessive max breadth leads to worse results, suggesting that incorporating too much information in the adaptive tree-based representation limits the LLM reasoning ability. This links back to the intuition of having an efficient representation for the LLM reasoning over long videos.

VLM/LLM Choice. We ablate the design choice of captioner and LLM for the VIDEOTREE framework in Tab. 6. The results show that with a better captioner and LLM, VIDEOTREE can achieve better performance on long video understanding tasks, indicating the potential VIDEOTREE to improve as its modules become more advanced. Notably, our GPT-3.5 variant substantially outperforms

Table 5: The effect of different settings for the max breadth of the first level of the tree. Results show that when the max breadth is set to 32, VIDEOTREE obtains the best performance. Reducing the max breadth can improve efficiency while retaining performance.

Table 6: The effect of different design choices of the captioner and LLM for VIDEOTREE on EgoSchema subset. The results show that with better LLM and a suitable captioner, our VIDEOTREE framework can boost the long video reasoning ability of the LLM.

Max Breadth	ES Acc	#Frame	Captioner	LLM	ES Acc	
8	62.0	36.9	BLIP-2 [24]	GPT-3.5	50.8	
16	64.4	49.0	LaViLA [85]	GPT-3.5	57.6	
32	66.2	63.2	BLIP-2 [24]	GPT-4	58.8	
64	62.4	94.6	LaViLA [85]	GPT-4	66.2	

existing methods with the same LLM (VideoAgent [62] 48.8%, LLoVi [83] 51.8%), achieving 57.6% accuracy on EgoSchema subset.

Error Bar Analysis. We provide the error bar analysis for our main results in Tab. 7.

Table 7: Comparison with other methods on EgoSchema, NExT-QA, and IntentQA datasets with error bar analysis.

Model	LLM		EgoSchema		IntentQA						
		Tem.	Cau.	Des.	Avg.	Sub.	Full	Full			
Video Transformer Models											
LongViViT [44]	-	-	-	-	-	56.8	33.3	-			
MC-ViT-L [2]	-	-	-	-	65.0	62.6	44.4	-			
InternVideo [63]	-	43.4	48.0	65.1	49.1	32.1	-	-			
InternVideo2 [64]	-	-	-	-	-	41.1	-	-			
Based on open-source LLMs											
SeViLA [79]	FlanT5-3B	61.3	61.5	75.6	63.6	25.7	22.7	60.9			
MVU [47]	Mistral-13B	55.4	48.1	64.1	55.2	60.3	37.6	-			
Based on proprietary LLMs											
ProViQ [9]	GPT-3.5	-	-	-	64.6	57.1	-	-			
LLoVi [83]	GPT-4	61.0	69.5	75.6	67.7	57.6	50.3	64.0			
VideoAgent [62]	GPT-4	64.5	72.7	81.1	71.3	60.2	54.1	-			
VideoAgent [11]	GPT-4	-	-	-	-	62.8	-	-			
MoReVQA [41]	PaLM-2	64.6	70.2	-	69.2	-	51.7	-			
IG-VLM [20]	GPT-4V	63.6	69.8	74.7	68.6	59.8	-	64.2			
VIDEOTREE (Ours)	GPT-4	67.0 (±0.5)	75.2 (±0.4)	81.3 (±0.8)	73.5 (±0.4)	66.2 (±1.2) 61.1	66.9 (±0.6)			

B Detailed Algorithm

In Algorithm 1, we present the algorithm behind VIDEOTREE.

C Additional Implementation Details

Prompt Details. We provide detailed prompts for the relevance scoring prompt in Tab. 8 and LLM reasoning prompt in Tab. 9 on the EgoSchema benchmark.

Experiments Compute Resources. All experiments are conducted on 4 (or less) NVIDIA-A6000 GPU and Azure Cloud APIs (for OpenAI models). The minimal GPU memory requirement is 24GB.

Algorithm 1 VIDEOTREE

Require:

```
V: Video frames from 1 to T
    Q: Video query
    k_init: Initial number of clusters
    relevance threshold: Minimum relevance score for significant clusters
    min_relevance: Minimum number of relevant clusters required
    max k: Maximum number of clusters allowed
    w: Branch width of the tree structure
 1:
 2: function ClusterAndAdd(parent, relevance)
 3:
        if relevance \leq 1 then
 4:
            return // max depth reached
        end if
 5:
        subcluster, subassignment \leftarrow VideoClustering(parent.frames, w) // cluster the parent
 6:
    node frames into two clusters
        for j \in \{1, \ldots, |subcluster|\} do
 7:
            cluster \leftarrow subcluster[j]
 8:
            added node \leftarrow tree.add node(parent.name, cluster, relevance) // add a node with
 9:
    parent, node content, and relevance
            ClusterAndAdd(added node, relevance -1) // recursively repeat for each child node
10:
        end for
11:
12: end function
13: function VIDEOTREE(V, k init, relevance threshold, min relevance, max k)
14:
        tree \leftarrow \text{Tree}()
        tree.add node("root", \emptyset, -1) // start with root node
15:
        k \leftarrow k\_init
16:
        clustering, assignment \leftarrow VideoClustering(V, k) // initial clustering into k clusters
17:
        relevance \leftarrow RelevanceScore(clustering)
18:
        n_{relevant} \leftarrow |\{x \ \forall x \in relevance : x > relevance_{threshold}\}| // get number of clusters
19:
    over threshold
        // breadth-wise expansion
20:
21:
        while n_{relevant} < min_{relevance} \& k \le max_k do
            k \leftarrow k * 2 // double the number of clusters until we have enough over the threshold
22:
            clustering, assignment \leftarrow VideoClustering(V, k)
23:
            relevance \leftarrow RelevanceScore(clustering)
24:
            n\_relevant \leftarrow |\{x \ \forall x \in relevance : x > relevance\_threshold\}|
25:
        end while
26:
27:
        for i \in \{1, \ldots, |clustering|\} do
28:
            cluster, rval \leftarrow clustering[i], relevance[i]
29:
            tree.add_node("root", cluster, rval) // add each cluster as a node with root as its parent
30:
        end for
31:
        // iterate over top-level clusters and descend depth-wise
32:
        for node in tree.nodes() do
            if node.rval \leq 1 then
33:
                continue
34:
35:
            end if
36:
            ClusterAndAdd(node, node.relevance) // descend into each node
37:
        end for
        frames\_to\_caption \leftarrow tree.get\_all\_nodes()
38:
39:
        captions \leftarrow GetCaptions(frames\_to\_caption)
40:
        pred\_answer \leftarrow LLMReasoning(captions, query)
        return pred_answer
41:
42: end function
```

Table 8: VIDEOTREE with relevance scoring prompt on EgoSchema.

User

You are presented with a textual description of a first view video clip, it consists of about caption_number frame captions sparsely sampled from the video (#C means the first person view, and #O indicates another). The ultimate goal is to answer a question related to this video, choosing the correct option out of five possible answers.

It is crucial that you imagine the visual scene as vividly as possible to enhance the accuracy of your response. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence. Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct. Meanwhile, could you provide a relevance score for each frame caption to evaluate their relevance with the query-answering process. The score is between 1,2,3, where 1 indicates low relevance and 3 signifies high relevance. Please return the relevance score in the format of a list of caption_number scores.

Examples: Examples Description: Captions Question: Question Options: A: Option-A. B: Option-B. C: Option-C. D: Option-D. E: Option-E. The prediction, explanation, confidence and frame relevance are (please response in the format of 'prediction:, explanation:, confidence:, frame relevance:')

Assistant prediction, explanation, confidence, frame relevance

D Additional Qualitative Analysis

In Fig. 5 we show another visualization from VIDEOTREE. Here, VIDEOTREE localizes a single key activity (embroidering a cloth) taking place in the video and dynamically expands its constituent frames to answer the question correctly using a minimal number of frames.

E License

We will make our code and models publicly accessible. We use standard licenses from the community and provide the following links to the licenses for the datasets, codes, and models that we used in this paper.

LLoVi: MIT LifelongMemory: MIT NExT-QA: MIT IntentQA: IntentQA EgoSchema: Ego4D license Kmeans-pytorch: MIT

PyTorch: BSD-style

Huggingface Transformers: Apache

Table 9: VIDEOTREE with LLM reasoning prompt on EgoSchema.

User

You are presented with a textual description of a first view video clip, it consists of frame captions sparsely sampled from the video (#C means the first person view, and #O indicates another). The ultimate goal is to answer a question related to this video, choosing the correct option out of five possible answers.

It is crucial that you imagine the visual scene as vividly as possible to enhance the accuracy of your response. After selecting your answer, rate your confidence level in this choice on a scale from 1 to 100, where 1 indicates low confidence and 100 signifies high confidence. Please provide a concise one-sentence explanation for your chosen answer. If you are uncertain about the correct option, select the one that seems closest to being correct.

Examples: Examples Description: Captions

Question: Question

Options: A: Option-A. B: Option-B. C: Option-C. D: Option-D. E: Option-E.

The prediction, explanation, and confidence is (please response in the format of 'prediction:, explanation: ,confidence:')

Assistant

prediction, explanation, confidence

Torchvision: BSD 3-Clause

SKLearn: BSD 3-Clause



Figure 5: Qualitative examples of VIDEOTREE keyframes and captions selection. Red options are answered wrongly with uniformly sampled frames. Green options are answered correctly by VIDEOTREE. Best viewed in color.