Video Summarization with Large Language Models

Min Jung Lee[1,2] Dayoung Gong[1] Minsu Cho[1]

[1]Pohang University of Science and Technology (POSTECH) [2]GenGenAI

Abstract

*The exponential increase in video content poses significant challenges in terms of efficient navigation, search, and retrieval, thus requiring advanced video summarization techniques. Existing video summarization methods, which heavily rely on visual features and temporal dynamics, often fail to capture the semantics of video content, resulting in incomplete or incoherent summaries. To tackle the challenge, we propose a new video summarization framework that leverages the capabilities of recent Large Language Models (LLMs), expecting that the knowledge learned from massive data enables LLMs to evaluate video frames in a manner that better aligns with diverse semantics and human judgments, effectively addressing the inherent subjectivity in defining keyframes. Our method, dubbed **LLM**-based **V**ideo **S**ummarization (**LLMVS**), translates video frames into a sequence of captions using a Muti-modal Large Language Model (M-LLM) and then assesses the importance of each frame using an LLM, based on the captions in its local context. These local importance scores are refined through a global attention mechanism in the entire context of video captions, ensuring that our summaries effectively reflect both the details and the overarching narrative. Our experimental results demonstrate the superiority of the proposed method over existing ones in standard benchmarks, highlighting the potential of LLMs in the processing of multimedia content.*

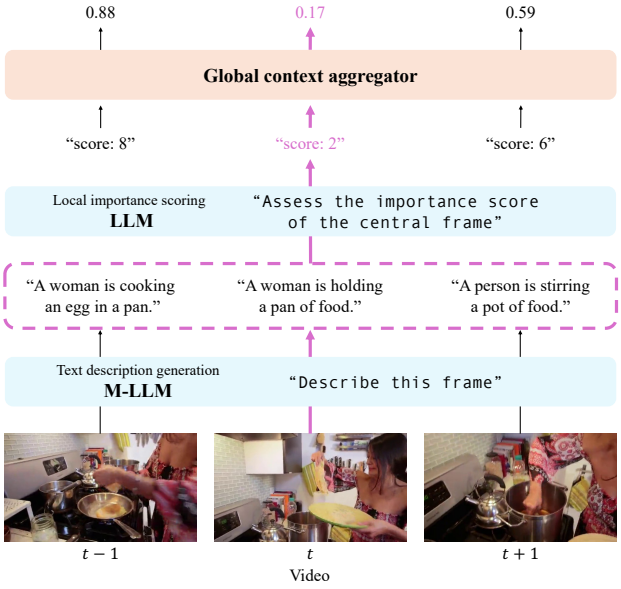


Figure 1. **Video summarization with (M-)LLMs.** Given the input video frames, captions for each frame are generated using M-LLM. For each frame at time-step t, the generated captions within a local window are grouped and provided as input to the LLM. The LLM is prompted to assess the importance score of the frame at time step t by considering this local context. Finally, a global context aggregator produces the final predictions by taking into account the overall context of the entire video. Note that, in this illustration, the local window size is set to 3.

1. Introduction

Video summarization is essential in multimedia content processing, particularly as the exponential growth in video data has far exceeded human capacity for consumption. Every day, millions of videos are uploaded across platforms, posing significant challenges in efficient navigation, search, and retrieval of video content. Video summarization addresses these challenges by condensing lengthy videos into concise summaries that capture the essential content. In response, researchers have explored automatic video summarization techniques aimed at producing videos that are shorter, more digestible, and appealing to users. However, summarizing video content remains complex due to its varied nature and the subjective elements of effective summarization.

Previous video summarization methods [5, 11, 12, 18, 35, 59] have primarily focused on selecting important frames solely based on visual features. Recent multi-modal methods [6, 14, 22, 28] integrate both visual and language modalities to leverage the contextual richness of natural language. However, these methods still prioritize visual features, incorporating textual data via an attention mechanism [44], where visual features serve as queries and language features as keys and values. While textual data helps to enhance the visual features, the main focus of video summarization still remains on visual content.

The advent of Large Language Models (LLMs) [1, 3, 4, 42, 43] presents new opportunities for video summarization. LLMs have shown strong capabilities in contextual understanding [7, 20], cross-domain reasoning [21, 48], and multimodal processing [2, 8, 10, 23, 25], allowing them to identify key moments based on semantic insights rather than visual saliency alone. Leveraging these strengths, we introduce LLMVS, an LLM-based video summarization framework that utilizes LLMs as important frame selectors, guided by textual data and embedded knowledge.

To this end, we propose a local-to-global video summarization model, as illustrated in Figure 1. First, we obtain textual data for each frame by generating textual descriptions from video frames using a pre-trained multi-modal LLM (M-LLM) [25]. Textual descriptions of video frames within a local window are fed into the LLM [43], along with structured instructions and examples in natural language, to perform in-context learning for video summarization. The LLM then evaluates the importance score of the center frame within the local context. Unlike existing methods that rely on the end output of LLMs [19, 34, 51, 57], our method extracts the output embeddings from LLMs and apply self-attention on them to aggregate global context from the videos and make the final predictions. During learning, the M-LLM and LLM are frozen to preserve their general domain knowledge, and only the self-attention blocks are trained.

Our contributions can be summarized as follows: **1)** We introduce LLMVS, a novel video summarization framework that leverages LLMs to utilize textual data and general knowledge in video summarization effectively. **2)** The proposed local-to-global video summarization framework integrates local context via window-based aggregation and global context through self-attention, enabling a comprehensive understanding of video content. **3)** Experimental results show that using output embeddings from LLMs is more effective for video summarization than using direct answers generated by LLMs. **4)** Comprehensive results demonstrate the effectiveness of the proposed method, achieving state-of-the-art performance on the SumMe and TVSum datasets.

2. Related Work

2.1. Video Summarization

Recent advancements in video summarization have significantly leveraged deep learning techniques by capturing temporal dynamics. A notable direction in this domain employs LSTMs [15, 17, 45, 50, 53–56] which are adept at capturing both short- and long-range dependencies in sequential frames. A pioneering work by Zhang et al.[53] utilizes Long Short-Term Memory (LSTM) networks, leveraging their ability to model variable temporal dependencies among video frames. Building on this foundation, subsequent studies explored various LSTM-based architectures for video summarization, such as hierarchical frameworks[54], stacked LSTMs [45], and encoder-decoder structures [17]. Transitioning to the utilization of self-attention mechanisms [5, 11, 12, 18, 47, 49, 59], VASNet [11] employs soft self-attention. Other approaches introduce localization components to guide attention, such as DSNet [59], which predicts the spatial offsets of interest regions, and iPTNet [18], which integrates moment localization through collaborative learning. Positional encoding has also been explored, as in PGL-SUM [5], which incorporates absolute position information into multi-head attention. CSTA [35] initially extracts and concatenates frame features, representing the temporal sequence as an image. This representation is then processed by a 2D CNN, yielding attention maps that capture spatiotemporal dependencies. These models primarily rely on visual cues and temporal features. In contrast, our work leverages the capabilities of LLMs to incorporate semantic information, enriching the contextual understanding of the video.

2.2. Multi-Modal Video Summarization

Unlike unimodal methods that rely solely on visual frames, multimodal video summarization [6, 14, 22, 28] integrates multiple modalities, such as visual and textual features, to produce more comprehensive summaries. CLIP-It [28] utilizes a cross-attention module between visual and textual features, both extracted using CLIP [32], allowing summarization to be conditioned on natural language. A2Summ [14] introduces an alignment-guided self-attention module that effectively fuses different modalities by leveraging the temporal correspondence between video and text, incorporating captions generated by GPT-2 [31] or existing transcript. Argaw et al.[6] propose a cross-modal attention mechanism to integrate multimodal cues from contextualized features, employing SRoBERTa-NLI-large[33] for sentence embedding and CLIP [32] for visual features. Prior multimodal video summarization methods [6, 14, 28] employ cross-attention mechanisms, where visual features act as queries and language features serve as keys and values. While these approaches incorporate language to enhance semantic understanding, they primarily focus on refining visual representations, often treating textual information as auxiliary information. In this paper, LLMVS leverages contextual understanding capabilities of LLMs for video summarization by utilizing both textual data and the general knowledge encoded in LLMs.

2.3. Video Understanding with LLM

Recent advancements in natural language processing (NLP) have been significantly driven by Large Language Models (LLMs)[1, 9, 39, 40, 42, 43]. The widespread adoption of these models spur the development of multimodal models that seamlessly integrate vision and text data[19, 27, 36,

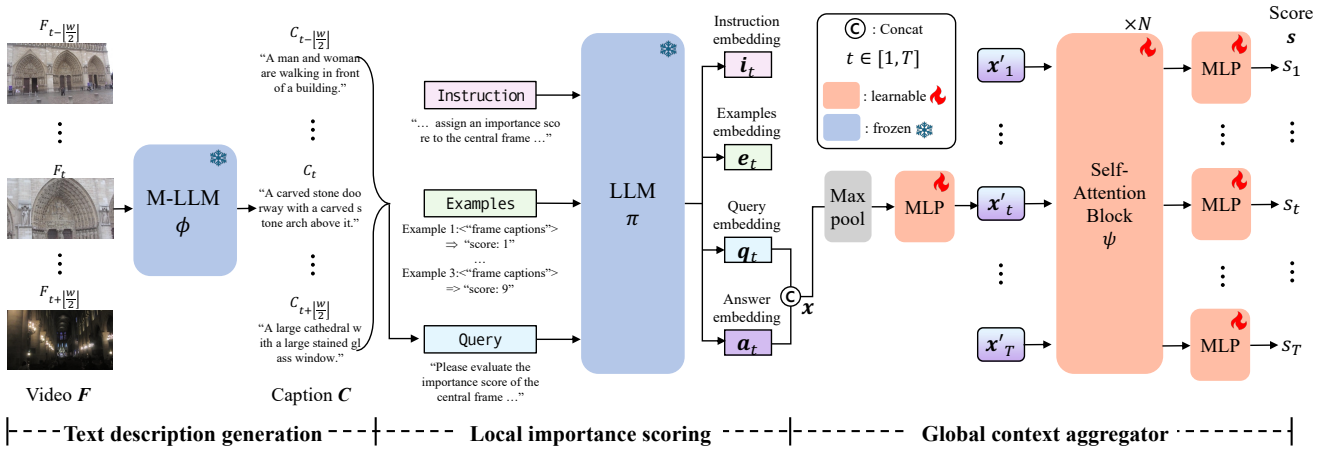


Figure 2. **Overall architecture.** Our LLMVS framework consists of three key components: text description generation, local importance scoring, and global context aggregation. First, captions for each video frame are generated using a pre-trained Multi-modal Large Language Model (M-LLM)[25]. These captions are then incorporated into the query component of an LLM [43] by segmenting through a sliding window local context, while instructions and examples are provided as part of the in-context learning prompt. We obtain the output embeddings from an intermediate layer of the LLM [43], categorized into instructions, examples, queries, and answers. The query and answer embeddings are pooled and passed through an MLP to produce inputs for the global context aggregator, which encodes the overall context of the input video. Finally, we obtain the output score vectors for the corresponding input video frames.

52, 57]. MovieChat [36] enhances video understanding by processing video representations with a Q-former [24] and a linear projection layer. These components convert visual features into text space before feeding them into a LLM for user interaction. In the realm of video question answering, MoReVQA [27] employs LLM in a multistage modular reasoning framework that breaks down complex queries into event parsing, grounding, and reasoning stages to interpret complex queries. Similarly, AntGPT [57] addresses action anticipation task by leveraging LLMs to infer pseudo-ground truth from observed actions and generate future steps. These works highlight the versatility of M-LLMs in merging data modalities and transforming interactions across domains. Inspired by these advancements, our approach applies M-LLM and LLM to the video summarization task, leveraging their ability to incorporate semantic information and provide a richer contextual understanding of video content.

3. LLM-based Video Summarization (LLMVS)

In this section, we present LLMVS, an LLM-based video summarization framework. Figure 2 shows the overall architecture of LLMVS. LLMVS consists of three parts: text description generation via M-LLM, local important frame scoring via LLM, and global context aggregation using self-attention blocks for final predictions of video summarization.

We begin by introducing the problem setup for video summarization in Section 3.1. Text description generation via M-LLM is discussed in Section 3.2. Local importance scoring, including in-context learning and extracting embeddings from LLM, is presented in Section 3.3. The global context aggregation using self-attention is detailed in Section 3.4, and the training objective is provided in Section 3.5.

3.1. Problem Setup

Given a video $\boldsymbol{F} = [F_1, F_2, \ldots, F_T] \in \mathbb{R}^{T \times H \times W \times 3}$, where T represents the temporal length of the video and H and W denote the height and width of each frame, respectively, the goal of video summarization is to obtain a sequence of importance scores $\boldsymbol{s} = [s_1, s_2, \ldots, s_T] \in \mathbb{R}^{T \times 1}$, where higher scores indicate more significant frames.

3.2. Text Description Generation via M-LLM

To incorporate textual data into video summarization, we first generate descriptions of input video frames using a pre-trained M-LLM, denoted by ϕ [25]. Specifically, we prompt ϕ with "Provide a detailed one-sentence description," generating a sequence of captions $\boldsymbol{C} = [C_1, C_2, \ldots, C_T]$,

$$\boldsymbol{C} = \phi(\boldsymbol{F}) \tag{1}$$

where C_i is the caption for the i-th frame.

3.3. Local Importance Scoring via LLM

Given a sequence of captions $\boldsymbol{C}$, we employ a pre-trained LLM π [43] to evaluate the importance of each frame within its local temporal context. Due to the inherent redundancy in video frames, it is essential to identify key frames based on local context rather than individual frames to filter out repetitive information. To achieve this, we apply a sliding window-based scoring method. Specifically, for each frame

[Instruction]	You are an intelligent chatbot designed to critically assess the importance of a central frame within a specific context . … Evaluate the frame using the following criteria: 1. Narrative Significance 2. Uniqueness and Novelty 3.Action and Dynamics
[Example 1]	Please evaluate the importance score of the central frame #7 in following 13 frames. #1 frame's caption … #13 frame's caption
[Answer]	score: 1
[Example 2]	Please evaluate the importance score of the central frame #3 in following 7 frames. #1 frame's caption … #7 frame's caption
[Answer]	score: 5
[Example 3]	Please evaluate the importance score of the central frame #6 in following 11 frames. #1 frame's caption … #11 frame's caption
[Answer]	score: 9
[Query]	Please evaluate the importance score of the central frame #$\lfloor w \rfloor$ in following w frames. #1 frame's caption … #w frame's caption

Figure 3. **In-context learning prompt of LLM.** The instruction for the LLM outlines the video summarization task and specifies the criteria. Then, three examples are provided. Each example includes the number of frame captions and identifies the central frame number as the target. In the query part, the frame captions of our focused video are passed.

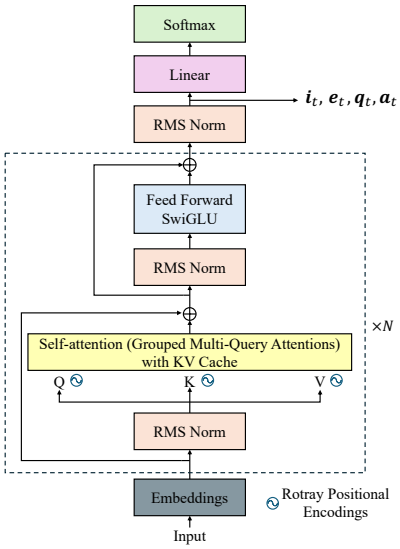


Figure 4. **Output embedding from LLM (Llama-2).** Among the output embeddings of instruction $\boldsymbol{i}_t$, examples $\boldsymbol{e}_t$, query $\boldsymbol{q}_t$, and answer $\boldsymbol{a}_t$ after the RMS Norm layer of the LLM (Llama-2), we utilize $\boldsymbol{q}$ and $\boldsymbol{a}$, which retains richer contextual and semantic information of the frame within a local context.

F_t at time-step t, the captions within a window of size w, denoted as $C_{t-\lfloor \frac{w}{2} \rfloor : t+\lfloor \frac{w}{2} \rfloor}$, are fed into the LLM π to evaluate the importance of the central frame C_t in relation to its surrounding frames. Here, $\lfloor . \rfloor$ denotes the floor function.

In-context learning for video summarization. To guide the LLM in generating task-specific outputs for video summarization, we apply in-context learning [36, 57], providing examples and instructions directly within the prompt, as shown in Figure 3. The prompt consists of three components: instructions, examples, and queries. The instructions define the frame scoring task and criteria; the examples provide three sample question-answer pairs related to video summarization; and the queries contain actual questions for the LLM to answer. The instructions and examples remain fixed, while only the queries vary based on the input. The full prompt configuration is provided in the Appendix B. In this way, we can effectively leverage the pre-trained LLM for video summarization without finetuning, enabling it to generate consistent and task-specific outputs based on the provided examples and instructions.

Output embeddings from LLM. Rather than obtaining direct answers from the LLM, we extract and utilize output embeddings in video summarization, which retain richer contextual and semantic information from the internal representations. This method offers a potentially more robust assessment of frame importance, preserving essential details that could be abstracted away in final answer outputs. Specifically, these embeddings are extracted from the Llama-2 [43] after the RMS Norm layer, as illustrated in Figure 4.

For each frame t, the LLM processes an in-context learning prompt consisting of instruction $\boldsymbol{i}$, examples $\boldsymbol{e}$, query $\boldsymbol{q}$, and answer $\boldsymbol{a}$. Since the instructions embeddings $\boldsymbol{i}$ and example embeddings $\boldsymbol{e}$ remain constant across frames, we focus on the window-specific query embeddings $\boldsymbol{q}$ and corresponding answer embedding $\boldsymbol{a}$. As a result, the query embeddings $\boldsymbol{q}_t \in \mathbb{R}^{L^{\mathrm{q}} \times D}$ and the answer embeddings $\boldsymbol{a}_t \in \mathbb{R}^{L^{\mathrm{a}} \times D}$ are obtained from the LLM π:

$$\boldsymbol{q}_t, \boldsymbol{a}_t = \pi(\boldsymbol{C}_{t-\lfloor \frac{w}{2} \rfloor : t+\lfloor \frac{w}{2} \rfloor}), \tag{2}$$

where L^{q} and L^{a} denote the sequence lengths of each embedding, respectively, and D represents the hidden dimension. Here, $\boldsymbol{q}_t$ and $\boldsymbol{a}_t$ encode the semantic information relevant to the frame at time-step t within its local context w.

3.4. Global Context Aggregating via Self-Attention

While the LLM effectively identifies important frames based on local context, incorporating global context is essential for producing a coherent summary of the entire video. To address this, we apply self-attention blocks ψ [44], enabling the model to capture dependencies across the entire video for the final important score prediction.

Within each local window centered at timestep t, we first concatenate $\boldsymbol{q}_t$ and $\boldsymbol{a}_t$ along the sequence axis, producing $\boldsymbol{x}_t \in \mathbb{R}^{(L^{\mathrm{q}}+L^{\mathrm{a}}) \times D}$:

$$\boldsymbol{x}_t = \texttt{concat}(\boldsymbol{q}_t, \boldsymbol{a}_t). \tag{3}$$

Then, max pooling is applied to $\boldsymbol{x}_t$ along the sequence axis, followed by an MLP, resulting in input embeddings $\boldsymbol{x}'_t \in \mathbb{R}^{1\times M}$:

$$\boldsymbol{x}'_t = \texttt{MLP}(\texttt{maxpool}(\boldsymbol{x}_t)). \tag{4}$$

The input embeddings for all timesteps, $\boldsymbol{x}' = [\boldsymbol{x}'_1, ..., \boldsymbol{x}'_T] \in \mathbb{R}^{T\times M}$, are provided to the global attention blocks ψ:

$$\boldsymbol{s} = \texttt{MLP}(\psi(\boldsymbol{x}')), \tag{5}$$

producing the final output of importance scores for the entire video, $\boldsymbol{s} \in \mathbb{R}^{T\times 1}$.

3.5. Training Objective

The proposed method is trained using the Mean Squared Error (MSE) loss to optimize frame importance predictions. The loss $\mathcal{L}$ between the ground truth score vector $\hat{\boldsymbol{s}}$ and the predicted score $\boldsymbol{s}$ and is defined as:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{s}_t - \hat{\boldsymbol{s}}_t)^2. \tag{6}$$

4. Experiment

4.1. Datasets

To evaluate the performance of our method, we use two well-known benchmarks: SumMe [13] and TVSum [37].

SumMe. The SumMe [13] comprises 25 videos, each ranging from 1 to 6 minutes in length, with an average duration of 2 minutes and 40 seconds. These videos, captured with egocentric, moving, and static cameras, cover various topics, including holidays, events, and sports. Each video is annotated by 15 to 18 raters.

TVSum. The TVSum [37] includes 50 videos with durations between 2 and 10 minutes, averaging 4 minutes and 11 seconds. This dataset spans diverse content types, such as how-to videos, documentaries, and vlogs. Each video is segmented into equal-length shots, and importance scores are assigned by 20 raters to these segments.

4.2. Evaluation Setup

The evaluation protocols for SumMe and TVSum differ in how ground truth and prediction are constructed. In SumMe, the ground truth summary is generated by averaging binary annotations from multiple users. Following the procedure in [53], we convert the predicted frame-level importance scores s into video summaries, by aggregating frame scores at the shot level using Kernel Temporal Segmentation (KTS) [30], which identifies shot boundaries. Shots with the highest importance scores are selected to form the summary, addressing the 0/1 knapsack problem and ensuring the summary length does not exceed 15% of the original total duration of the video. The resulting summary is evaluated against the ground truth summary to measure performance.

For TVSum, importance scores of each user on a continuous 0–1 scale serve as the ground truth. Evaluation is conducted by comparing the predicted scores s individually with annotations of each user, and the final performance is computed by averaging the results across users.

We evaluate our method using the standard 5-fold cross-validation protocol, following the previous approaches [5, 14, 18, 35, 59]. As evaluation metrics, we adopt rank order statistics, specifically Kendall's τ and Spearman's ρ following [29]. While the F1 score is widely used in video summarization, it favors short shots over key shots due to length constraints [29, 35, 41]. This limitation may result in an inaccurate reflection of summarization quality. We thus exclude it from our evaluation metrics.

4.3. Implementation Details

We employ LLaVA-1.5-7B [25] as the text description generator model ϕ, and Llama-2-13B-chat [43] as the local importance scoring model π. The length of each frame caption generated by ϕ is limited to a maximum of 77 tokens. We train our model using the AdamW optimizer [26] across 200 epochs on 5 NVIDIA A100 GPUs with a batch size of 1. The total training time is approximately 10 hours. The learning rate is set to $1.19e-4$ for the SumMe [13] and $7e-5$ for the TVSum [37]. For both datasets, the window size w is set to 7 and the dimension reduction M is 2048. The number of self-attention blocks and the number of heads for global context aggregator ψ are set to 3 and 2, respectively. The frame captions used in the example section for in-context learning are randomly sampled from the training set of SumMe.

4.4. Comparison with the State of the Art

Table 1 compares the performance of LLM and LLMVS with the state-of-the-art models on two benchmark datasets. The table is divided into three compartments: (1) random and human baselines, (2) models utilizing use *visual* features, and (3) models using both *visual* and *text* features. Random and human performance metrics are from [29], where random performance is computed by averaging the results of comparisons between 100 randomly generated value sequences in the range $[0, 1]$ and the ground truth.

LLMs. Table 1 investigates the impact of the general knowledge of LLM [43] on video summarization by evaluating it in a zero-shot setting. Importance scores are obtained via in-context learning (Figure 3), using the same experimental setup as LLMVS, but without the global context aggregator ψ. The LLM achieves competitive performance among previous methods on SumMe, demonstrating the effectiveness of leveraging textual data alongside its general knowledge. In contrast, its performance on TVSum is comparatively lower. This discrepancy can be attributed to differences in the evaluation protocols of SumMe and TVSum, as described in

Method	SumMe		TVSum	
	τ	ρ	τ	ρ
Random [29]	0.000	0.000	0.000	0.000
Human [29]	0.205	0.213	0.177	0.204
Visual				
VASNet [11]	0.160	0.170	0.160	0.170
DSNet-AB [59]	0.051	0.059	0.108	0.129
DSNet-AF [59]	0.037	0.046	0.113	0.138
DMASum [46]	0.063	0.089	0.203	0.267
PGL-SUM [5]	-	-	0.206	0.157
MSVA [12]	0.200	0.230	0.190	0.210
iPTNet [18]	0.101	0.119	0.134	0.163
CSTA [35]	0.246	0.274	0.194	0.255
Visual + Text				
CLIP-It [28]	-	-	0.108	0.147
A2Summ [14]	0.108	0.129	0.137	0.165
SSPVS [22]	0.192	0.257	0.181	0.238
Argaw *et al*. [6]	0.130	0.152	0.155	0.186
LLM	0.170	0.189	0.051	0.056
LLMVS (ours)	**0.253**	**0.282**	**0.211**	**0.275**

Table 1. **Comparison with the state of the arts.** The table is divided into three compartments: (1) random and human baselines, (2) models utilizing use visual features, and (3) models using both visual and text features. LLMVS achieves the state-of-the-art performance on two benchmark datasets.

Section 4.2. In SumMe, evaluation is performed by averaging the user summaries, whereas in TVSum, evaluation is conducted separately for each user score, and the results are then averaged. This performance gap, attributed to differences in subjectivity between the two datasets, indicates that the LLM is well-suited for general summarization tasks but less effective in capturing individual user preferences.

LLMVS. Our model achieves state-of-the-art performance on both benchmark datasets. Notably, LLMVS shows significant performance gains over the zero-shot LLM, as seen by comparing the last two rows in Table 1. The result indicates that the proposed method effectively handles both general and subjective aspects of keyframe selection in video summarization. In particular, it highlights the importance of the global context aggregator ψ, which enhances the reasoning ability of LLM by capturing contextual relationships across local windows, enabling more coherent and informed sequence-level decision-making. Moreover, our method outperforms existing multimodal video summarization models [6, 14, 22, 28], where text information serves as an auxiliary input to support visual processing. Unlike these approaches, our framework centers summarization around language, relying on textual descriptions and the reasoning capabilities of LLMs. These results underscore the advantage of integrating textual data with the broad reasoning capabilities of LLMs, enabling more contextually aware and semantically rich video summaries.

	ϕ	π	ψ	τ	ρ
(1)	-	LLaVA	-	0.119	0.132
(2)	-	LLaVA*	-	0.140	0.156
(3)	LLaVA	Llama	-	0.170	0.189
(4)	LLaVA	Llama*	-	0.181	0.201
(5)	**LLaVA**	**Llama**	**SAB***	**0.253**	**0.282**

Table 2. **Finetuning (M-)LLM, ϕ and π.** ϕ: text generator, π: local importance scorer, ψ: global context aggregator, *:finetuned, SAB: self-attention blocks.

Prompt type	τ	ρ
Central-background	0.241	0.269
Generic	**0.253**	**0.282**

(a) Prompting to M-LLM ϕ

Prompt type	τ	ρ
Textual explanation	0.239	0.266
Numerical evaluation	**0.253**	**0.282**

(b) Prompting to LLM π

Table 3. **Prompting to (M-)LLM.** Evaluation of different prompting styles applied to (a) M-LLM ϕ and (b) LLM π on the SumMe dataset [13]. All experiments use a window size of $w = 7$.

4.5. Analysis

Finetuning (M-)LLM, ϕ and π. To determine whether the performance improvements arise from fundamental architectural enhancements or are simply due to finetuning on downstream benchmarks, we conduct experiments on both zero-shot and finetuned settings using (M-)LLM.

We first establish a baseline by evaluating M-LLM (LLaVA [25]) and LLM (Llama-2 [43]) in zero-shot settings. In particular, we assess whether M-LLM can directly serve as the importance scorer π, assigning frame-level importance scores without relying on captions, as shown in the first row of Table 2. Comparing the first and third rows of the table demonstrates that explicitly providing captions from M-LLM to the LLM yields better results than direct scoring by M-LLM alone, underscoring the importance of leveraging language semantics in video summarization.

Subsequently, to evaluate the impact of finetuning, we apply LoRA [58] to both the M-LLM and LLM, with the finetuned models denoted by * in Table 2. The performance improvements observed when comparing the first and second rows, as well as the third and fourth rows, validate the effectiveness of finetuning. However, LLMVS exhibits significantly greater improvements in the fifth row compared to these baselines, demonstrating that its effectiveness extends beyond simple finetuning.

Prompting to (M-)LLM , ϕ and π. Prompting is essential in (M-)LLMs, as it determines how the model processes

	Query q	Answer a	ψ	τ	ρ
(1)	-	✓	SAB*	0.233	0.260
(2)	✓	-	SAB*	0.238	0.265
(3)	✓	✓	**SAB***	**0.253**	**0.282**
(4)	✓	✓	MLP*	0.182	0.203

Table 4. **Ablation studies.** The embeddings are used individually or concatenated. Performance is evaluated on the SumMe dataset [13]. *:finetuned, MLP: MLP only (without self-attention blocks), SAB: self-attention blocks.

information and generates responses. To evaluate the effectiveness of different prompting strategies, we examine prompts for both M-LLM ϕ and LLM π.

For M-LLM ϕ, we explore the impact of different captioning styles. Specifically, we examine how the richness and descriptiveness of frame captions influence summarization. As detailed in Section 3.2, we instruct the M-LLM with a generic prompt "Provide a detailed one-sentence description." To obtain more fine-grained descriptions, we instruct the model to separately describe the center and background regions of the image using two prompts: "Describe the center part of this image in one detailed sentence" and "Describe the background part of this image in one detailed sentence." Table 3 (a), the generic prompt yields better results than the central-background approach. This suggests that a broader, high-level description allows the model to better capture scene dynamics and key events, reducing reliance on specific frame regions.

For LLM π, we compare two prompting types: (1) direct numerical scoring of frame importance using the prompt, "Please evaluate the importance score of the central frame in following frames," as described in Figure 3; and (2) textual explanation, where the LLM is instructed to summarize frame captions within a local window using the prompt, "Please summarize the following frames in one sentence." inspired by [16]. Table 3 (b) shows that direct numerical scoring consistently outperforms textual summarization, suggesting that assigning explicit importance scores provides a clearer and more effective evaluation of frame significance.

Ablation studies. We conduct experiments to validate the effectiveness of using the output embeddings q and a and self-attention blocks ψ. Table 4 shows the results. From the first and second rows, we either use query embeddings q or answer embeddings a during input structuring in Section 3.4. Comparing the first and second rows shows that leveraging query embeddings yields better performance than leveraging answer embeddings alone, highlighting the importance of contextual information in assessing frame relevance and enriching the semantic processing capabilities of the LLM. Furthermore, the third row, which combines both query and answer embeddings with the global context aggregator, achieves the best results, confirming that integrating both query and answer embeddings with the global context aggregator yields the most effective results. In the fourth row, we replace the self-attention blocks (SAB) used as ψ with a simple MLP. A comparison between the third and fourth rows demonstrates that employing the global self-attention block is more effective than using an MLP.

Extraction position	τ	ρ
After Linear layer	0.241	0.269
After RMS Norm layer	**0.253**	**0.282**

Table 5. **Extraction position of output embeddings q and a.** Evaluation performed on SumMe with window size $w = 7$.

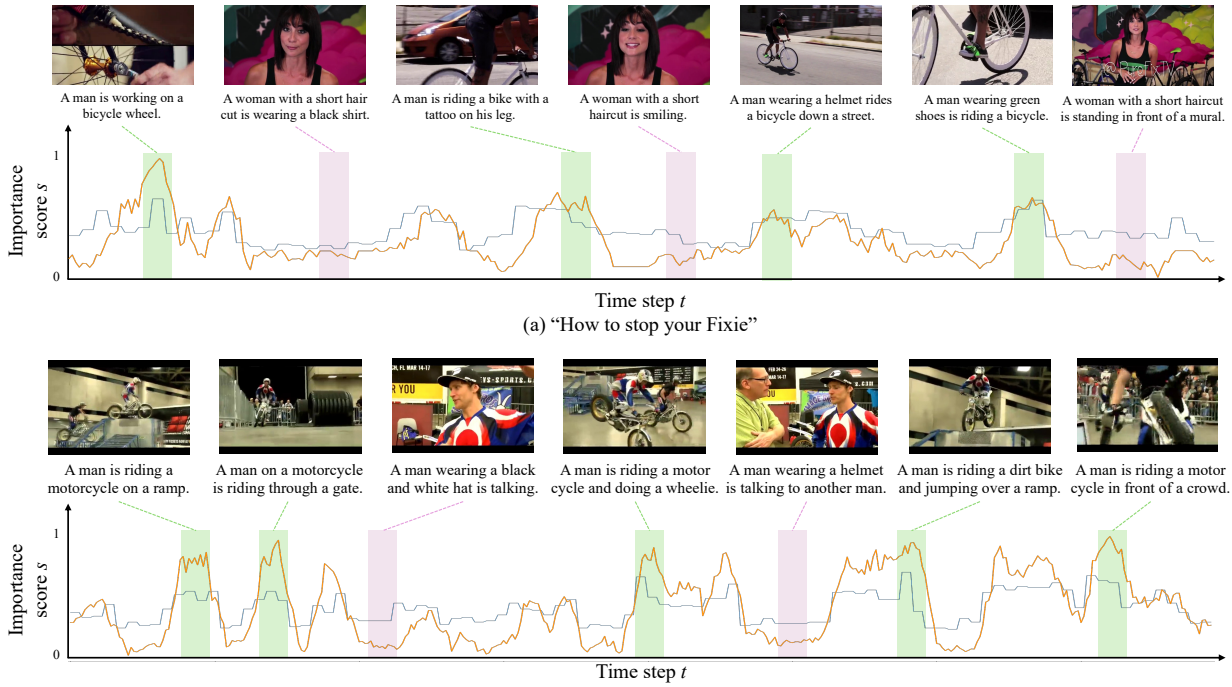
Method	τ	ρ
VASNet [44]	0.364	0.364
PGL-SUM [5]	0.375	0.375
DSNet-AB [59]	0.362	0.362
DSNet-AF [59]	0.342	0.342
LLMVS (ours)	**0.440**	**0.440**

Table 6. **Zeroshot evaluation on MR.HiSum.** The evaluation is conducted on 50 randomly selected videos from the MR.HiSum [38]. Both previous methods and LLMVS are tested directly on MR.HiSum using pre-trained models which are originally trained on SumMe.

Extraction position of output embeddings q and a. In Table 5, we examine the effects of extraction position of output embeddings q and a. Specifically, we consider two positions within the LLM π: after the RMS Norm layer and after the Linear layer, as shown in Figure 4. Since embeddings extracted after the linear layer are specialized for word domains, we aim to explore the effectiveness of embeddings obtained both before and after this specialization, namely after the RMS Norm layer and the Linear layer, respectively.

Table 5 presents that embeddings extracted after the RMS Norm layer outperform those after the Linear layer, likely due to their retention of richer contextual information, whereas embeddings after the Linear layer are more specialized for word domains.

Zero-shot evaluation on MR.HiSum. To evaluate the generalization ability of the proposed method on unseen videos, we train LLMVS on the SumMe [13] and test its zero-shot performance on a random subset of 50 videos from the MR.HiSum [38]. We compare LLMVS with previous methods [5, 44, 59]. Table 6 shows that LLMVS outperforms other models, demonstrating its strong generalization capability in zero-shot settings. This result suggests that by leveraging the advanced capability of the LLM to interpret text-based information and incorporating its contextual embeddings, LLMVS effectively captures representations that extend well to unseen video content.



(a) "How to stop your Fixie"

(b) "Smage Bros. Motorcycle Stunt Show"

Figure 5. **Qualitative results.** Videos are from the TVSum dataset [37]. The x-axis and y-axis represent time step t and importance score s, respectively. The blue line indicates the averaged user scores from the ground truth annotations, while the orange line denotes the predicted importance scores from our model, normalized to the range [0, 1]. Green regions highlight segments where importance scores are high, whereas pink regions indicate segments where importance scores are low.

4.6. Qualitative Results

Figure 5 presents qualitative results on TVSum [37], comparing predicted scores from LLMVS against the ground-truth user scores. The x- and y-axes are time step t and importance score s, respectively. In this figure, the blue line represents the average user scores, while the orange line shows the normalized predicted scores from our model. All scores are in the range of 0 to 1. Green areas indicate segments that received high importance scores, while pink areas correspond to segments with low scores.

The predicted importance scores align well with overall trends in the ground truth, highlighting the robustness and generalization capability of our approach. We also observe that action-related scenes tend to receive higher importance scores from both human annotators and LLMVS. For example, in Figure 5 (a), which presents a video about instructing how to stop a bike, scenes where a woman is talking to the camera receive relatively low scores. In contrast, frames depicting dynamic actions—such as riding or touching the bike—are assigned higher scores. Similarly, in Figure 5 (b), which features a motorcycle stunt show, frames showing a man being interviewed are rated lower, whereas those involving high-energy activities, such as stunts, receive higher scores. These patterns suggest that LLMVS effectively identifies and emphasizes action-oriented content that contributes significantly to the narrative.

5. Conclusion

We have introduced the LLM-based Video Summarization framework (LLMVS), which leverages the semantic understanding capabilities of large language models to perform video summarization through caption-guided frame scoring. LLMVS integrates textual descriptions generated by M-LLM from video frames, which are then evaluated and refined through the LLM using a comprehensive local-global context aggregation network. This design allows the model to capture narrative structure more effectively by combining the descriptive strength of the M-LLM with the reasoning capabilities of the LLM. Experiments on the SumMe and TVSum demonstrate the effectiveness of the proposed approach, showing consistent improvements over existing methods. By bridging the gap between visual data and language, LLMVS enhances the summarization process and sets a new direction for future research in multimedia content analysis, enabling advanced cross-modal reasoning.

Acknowledgements. This work was supported by the NRF grant (RS-2021-NR059830 (35%)), the IITP grants (RS-2022-II220264: Comprehensive Video Understanding and Generation (40%), RS-2019-II191906: Artificial Intelligence Graduate School Program at POSTECH (5%)) funded by the Korea government (MSIT), the DIPS1000+ grant (20266807: Generative AI Core Technology for Creating High-Quality Synthetic Data and Its Application to Vision AI System for Autonomous Driving) by KISED (10%), and the Defense Innovation Company 100 Exclusive R&D Support Program (R240202: Development of Sensor-Level High-Quality Multimodal/Video Synthesis Data Generation and Real-time Target Recognition Technology for Unmanned System) by the Korea goverment (KRIT) (10%).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[3] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023. 2

[4] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 2

[5] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *2021 IEEE international symposium on multimedia (ISM)*, pages 226–234. IEEE, 2021. 1, 2, 5, 6, 7

[6] Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Dernoncourt, and Joon Son Chung. Scaling up video summarization pretraining with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8332–8341, 2024. 1, 2, 6

[7] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[8] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040, 2022. 2

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023. 2

[10] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 2

[11] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*, pages 39–54. Springer, 2019. 1, 2, 6

[12] Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Supervised video summarization via multiple feature sets with parallel attention. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6s. IEEE, 2021. 1, 2, 6

[13] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 505–520. Springer, 2014. 5, 6, 7

[14] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14867–14878, 2023. 1, 2, 5, 6

[15] Tanveer Hussain, Khan Muhammad, Amin Ullah, Zehong Cao, Sung Wook Baik, and Victor Hugo C De Albuquerque. Cloud-assisted multiview video summarization using cnn and bidirectional lstm. *IEEE Transactions on Industrial Informatics*, 16(1):77–86, 2019. 2

[16] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18198–18208, 2024. 7

[17] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1709–1717, 2019. 2

[18] Hao Jiang and Yadong Mu. Joint video summarization and moment localization by cross-task sample transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16388–16398, 2022. 1, 2, 5, 6

[19] Apoorv Khandelwal, Ellie Pavlick, and Chen Sun. Analyzing modular approaches for visual question decomposition. *arXiv preprint arXiv:2311.06411*, 2023. 2

[20] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020. 2

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022. 2

[22] Haopeng Li, Qiuhong Ke, Mingming Gong, and Tom Drummond. Progressive video summarization via multimodal self-supervised learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5584–5593, 2023. 1, 2, 6

[23] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 2

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 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, 2023. 3

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2, 3, 5, 6

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[27] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13235–13245, 2024. 2, 3

[28] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *Advances in neural information processing systems*, 34:13988–14000, 2021. 1, 2, 6

[29] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7596–7604, 2019. 5, 6

[30] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 540–555. Springer, 2014. 5

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[33] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 2

[34] Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, David Seunghyun Yoon, Ryan A Rossi, and Franck Dernoncourt. Pdftriage: Question answering over long, structured documents. *arXiv preprint arXiv:2309.08872*, 2023. 2

[35] Jaewon Son, Jaehun Park, and Kwangsu Kim. Csta: Cnn-based spatiotemporal attention for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18847–18856, 2024. 1, 2, 5, 6

[36] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 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, 2024. 2, 3, 4

[37] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015. 5, 8

[38] Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. hisum: A large-scale dataset for video highlight detection and summarization. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[39] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023. 2

[40] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2

[41] Hacene Terbouche, Maryan Morel, Mariano Rodriguez, and Alice Othmani. Multi-annotation attention model for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3143–3152, 2023. 5

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[43] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2, 3, 4, 5, 6

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 4, 7

[45] Junbo Wang, Wei Wang, Zhiyong Wang, Liang Wang, Dagan Feng, and Tieniu Tan. Stacked memory network for video summarization. In *Proceedings of the 27th ACM international conference on multimedia*, pages 836–844, 2019. 2

[46] Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, and Xiaolin Wei. Query twice: Dual mixture attention meta learning for video summarization. In *Proceedings of the 28th ACM international conference on multimedia*, pages 4023–4031, 2020. 6

[47] Lezi Wang, Dong Liu, Rohit Puri, and Dimitris N Metaxas. Learning trailer moments in full-length movies with co-contrastive attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 300–316. Springer, 2020. 2

[48] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2

[49] Minghao Xu, Hang Wang, Bingbing Ni, Riheng Zhu, Zhenbang Sun, and Changhu Wang. Cross-category video highlight detection via set-based learning. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 7970–7979, 2021. 2

[50] Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9143–9150, 2019. 2

[51] Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. Harnessing large language models for training-free video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18527–18536, 2024. 2

[52] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

[53] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 766–782. Springer, 2016. 2, 5

[54] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7405–7414, 2018. 2

[55] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Tth-rnn: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Transactions on Industrial Electronics*, 68(4): 3629–3637, 2020.

[56] Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. Reconstructive sequence-graph network for video summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2793–2801, 2021. 2

[57] Qi Zhao, Ce Zhang, Shijie Wang, Changcheng Fu, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos? *arXiv preprint arXiv:2307.16368*, 2023. 2, 3, 4

[58] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024. 6

[59] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 30:948–962, 2020. 1, 2, 5, 6, 7