Scaling Up Video Summarization Pretraining with Large Language Models

Dawit Mureja Argaw[1,2] Seunghyun Yoon[2] Fabian Caba Heilbron[2] Hanieh Deilamsalehy[2]
Trung Bui[2] Zhaowen Wang[2] Franck Dernoncourt[2] Joon Son Chung[1]

[1] KAIST [2] Adobe Research

Abstract

Long-form video content constitutes a significant portion of internet traffic, making automated video summarization an essential research problem. However, existing video summarization datasets are notably limited in their size, constraining the effectiveness of state-of-the-art methods for generalization. Our work aims to overcome this limitation by capitalizing on the abundance of long-form videos with dense speech-to-video alignment and the remarkable capabilities of recent large language models (LLMs) in summarizing long text. We introduce an automated and scalable pipeline for generating a large-scale video summarization dataset using LLMs as Oracle summarizers. By leveraging the generated dataset, we analyze the limitations of existing approaches and propose a new video summarization model that effectively addresses them. To facilitate further research in the field, our work also presents a new benchmark dataset that contains 1200 long videos each with high-quality summaries annotated by professionals. Extensive experiments clearly indicate that our proposed approach sets a new state-of-the-art in video summarization across several benchmarks.

1. Introduction

In the current era of information, long-form video content constitutes a significant portion of internet traffic. Consequently, developing models for automated *video summarization* has become an essential research topic [5, 9, 10, 14, 20, 25, 35, 45–47, 49]. Video summarization involves automatically creating a condensed summary video from a longer input video, highlighting the key information. This task is highly practical as it allows users to selectively filter the content they wish to explore in greater detail (*e.g.* promotional trailers) or obtain concise summaries of the content they intend to consume (*e.g.* recap videos).

Learning to summarize videos, however, is a very ill-posed problem. This is mainly because of the diverse nature of video content and the subjective nature of what constitutes a meaningful summary. Therefore, an intuitive, data-driven approach to developing a *robust* video summarizer would involve exposing the model to a large set of video-summary pairs during training. However, obtaining such a dataset is a daunting and resource-intensive task, primarily due to the manual labor required for annotating summary videos. This challenge is reflected in existing video summarization datasets like TVSum [36] and SumMe [7] which are characterized by a notably small number of video-summary pairs, with only 50 and 25 pairs, respectively. Consequently, state-of-the-art video summarization methods [8, 9, 25, 26] tend to overfit to a specific video domain and their ability to generalize effectively is significantly limited.

The main focus of this work is to address these limitations. Motivated by the abundance of long-form videos with dense *speech-to-video* alignment [23] and the recent achievements of large language models (LLMs) [22, 27, 38] in comprehending and summarizing extensive textual content, we propose an *automatic* and *scalable* pipeline for large-scale video summarization pretraining. Our key idea is to leverage LLMs as Oracle summarizers to transfer their capabilities from text to the video domain. This enables the scaling up of visual summarization datasets, facilitating the training of video summarizers for scenarios where narration or text is not available.

Given a long narrated video, we first use a speech-to-text model [2, 32] to obtain the textual transcription of the video. Next, we input the text into the LLM in a format where each sentence in the transcript is accompanied by its corresponding timestamp. We then *prompt* the LLM to output an extractive summary of the video transcript by selecting only the most critical and informative moments from the video while maintaining the timestamp and original wording of the selected sentences. The main reason for providing this particular instruction is to guarantee that the extracted textual summary can be seamlessly associated with the corresponding video segments. Finally, we carefully map the extracted textual summary back to the relevant video segments. This process results in a sequence of clips that, when aggregated, form a pseudo-ground truth visual summary. Following this methodology, we create a large-scale

dataset, named *Long-form Video Summarization Pretraining* (**LfVS-P**) dataset, consisting of 250K video-summary pairs for training a robust video summarization model.

Leveraging the extensive dataset we have generated, our work conducts an analysis of various video summarization baselines. A common approach in most existing works is to frame video summarization as a binary classification [25] where each moment is classified as summary or not, or as frame importance prediction [8, 9, 26], estimating the likelihood of each frame being part of a summary. However, these approaches present two key limitations. Firstly, they suffer from a long-tail distribution problem, characterized by a significant class imbalance, as the number of summary moments in a video is considerably smaller compared to non-summary moments. Secondly, the decision of whether a video segment at a given time step is a summary or not happens independently, without consideration of what was previously classified as a summary, as predictions are made in parallel. This eventually leads to numerous repetitive moments being categorized as summary.

To address these limitations, our work proposes a new video summarization model. We adopt a regression-based approach in which the model decodes continuous feature representations of the summary moments, as opposed to predicting discrete binary classes or importance scores, in order to mitigate the long-tail distribution problem. Furthermore, we utilize an autoregressive decoding process, where the decoding at a given time step t is conditioned on the summary moments decoded up to time $t-1$ and the input video. This sequential scheme enables the network to learn intricate contextual dependencies between summary moments during the generation of a summary.

We design a Transformer-based [39] encoder-decoder architecture that takes a long video as input and autoregressively generates a short summary video. We approach video summarization as a multi-modal problem integrating both visual and textual (transcribed speech) cues, from the input video to guide the prediction of summary videos. Recognizing the prevalence of videos without narration or language cues, we train our framework to depend solely on visual cues when textual information is absent. Consequently, during inference, our model is versatile and can be deployed on videos, whether or not they come with accompanying text.

We conduct comprehensive experiments covering aspects such as problem formulation, network design, and scaling effects. Our results clearly indicate the benefits of large-scale pretraining using the collected dataset for robust cross-dataset generalization. Furthermore, to assess the effectiveness of video summarization models and to encourage ongoing research in this field, we introduce a new benchmark known as *Long-form Video Summarization Testing* (**LfVS-T**). This benchmark comprises a collection of 1,200 diverse videos, each paired with carefully annotated ground truth summaries produced by professional human annotators. We evaluate our model and existing approaches [8, 10, 25, 26] using various metrics. Our autoregressive approach outperforms previous works, establishing a new state-of-the-art across multiple benchmarks.

Contributions. Our work brings three main contributions to video summarization research: **(1)** We introduce an automatic and scalable mechanism that leverages publicly available long-form videos and LLMs as oracle summarizers to curate the LfVS-P dataset for large-scale video summarization pretraining. **(2)** We present a new video summarization model that effectively addresses the limitations of previous works and achieves state-of-the-art performance across several benchmarks. **(3)** To facilitate further research in the field, we introduce a new benchmark dataset named LfVS-T which contains 1,200 publicly available long videos with high-quality summaries annotated by humans.

2. Related Works

Text Summarization. Text summarization is a fundamental NLP task that aims to generate concise and informative summaries of texts [1, 6]. Extractive summarization extracts important sentences from the original text [17, 24], while abstractive summarization generates new summaries that convey the main points [15, 41]. Early works on extractive summarization include TextRank [24], an unsupervised graph-based algorithm. Liu *et al.* [17] enhanced it with contextualized word embeddings [17]. The advent of Transformers [39] has greatly advanced abstractive summarization, exemplified by models like BART [15] and Pegasus [41]. Recent progress in text summarization has been driven by the development of large language models (LLMs) [22, 27, 38]. These models are able to learn the semantics of the original text and generate summaries that are more informative and comprehensive than traditional summarization models [40, 44]. Additionally, LLMs have been shown to be capable of generating summaries in a variety of different formats, such as bullet points, paragraphs, and even code [29, 48]. Our work leverages the power of LLMs to create an extensive dataset for video summarization.

Video Summarization. Video summarization is the task of generating a concise representation of a video that captures the main events and ideas. Existing approaches can be broadly categorized as supervised and unsupervised. Many early works focused on unsupervised video summarization [5, 11, 14, 20, 21, 45] partly due to a lack of labeled training datasets. With the emergence of video summarization benchmarks such as SumMe [7] and TVSum [36] several supervised approaches [9, 10, 25, 35, 42, 43, 46, 47, 49] have been proposed. Most works focus on generic video summarization [9, 10, 46] where the most informative moments of an input video are temporally aggregated to com-

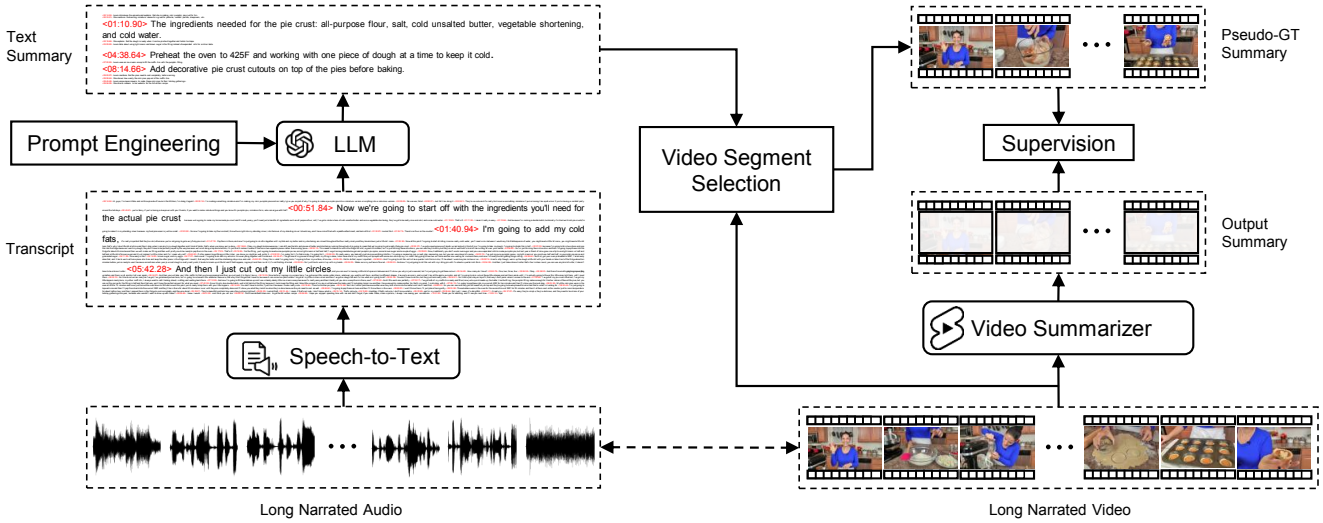

Figure 1. **Scalable Dataset for Video Summarization**. Given a long-form video with dense speech-to-video alignment, we first use a speech-to-text model [2] to transcribe the video. Next, we preprocess the text so that each sentence in the transcript is accompanied by its corresponding start timestamp. We then prompt an LLM [27, 38] to extract the most critical and informative moments from the video along with their timestamp. After extracting the textual summary, we map it back to the relevant video segments to compose a pseudo-ground truth summary. Following this pipeline, we generate a large-scale dataset of video-summary pairs for video summarization pretraining.

pose a summary video. Few other works have explored query-based summarization [12, 25, 34, 35], where user-defined natural language queries are used to customize the summaries. Other works have explored a multi-modal setup [8, 16, 26, 30] where a text input in the form of video captions or transcribed speech was incorporated along with the video input to guide video summarization. Our work follows a similar formulation and proposes a new video summarization model that attempts to mitigate the limitations of previous approaches.

3. Scalable Dataset for Video Summarization

Text summarization has undergone significant advancements in recent years, driven by the exceptional capabilities of large language models (LLMs) [22, 27, 38] in comprehending large textual content. In contrast, progress in video summarization has been notably constrained, primarily due to the challenge of obtaining a substantial annotated dataset for the task. Our work aims to bridge this gap by harnessing the power of LLMs to generate a scalable dataset for visual summarization pretraining. The overview of our proposed approach is shown in Fig. 1.

Source Data. Generating a precise pseudo-ground truth summary from a given video using LLMs as oracle summarizers hinges on the presence of a strong speech-to-visual alignment within the video. For this reason, we make use of the HowTo100M dataset [23] which contains more than 1.2M narrated web videos. Given that our work focuses on summarizing long-form videos, we only select videos that are 8 minutes or longer in duration. We then use a state-of-

I am providing you with a transcribed narration from a video, complete with timestamps. Please generate an extractive summary from this text. Here are your instructions:

1. The summary should consist of only the most critical and informative moments from the video.
2. Do not paraphrase or reword the sentences. Maintain their original wording.
3. Each sentence you extract for the summary must include its original timestamp.

<Transcript>

Figure 2. **Prompt Engineering**. We formulate a prompt instructing an LLM to perform an *extractive text summarization task*. We explicitly emphasize not paraphrasing the wording in the extracted sentences and retaining their timestamps. This ensures seamless matching of the text summary back to the input video.

the-art speech-to-text model, Whisper [2, 32], to transcribe the video. To address potential issues with noisy data curation, we use CLIP embeddings [31] to measure the similarity between the video frames and their corresponding text in the transcript. We subsequently remove videos where the narration lacks sufficient alignment with the visual content in the video.

Prompting LLMs for Extractive Text Summarization. After obtaining the transcript of a video, we perform text summarization using highly capable LLMs with a large context size [27, 38]. We first preprocess the transcript into a format where each sentence in the text is preceded by its start timestamp, as depicted in Fig. 1. This step provides temporal context to the text corpus when feeding it into the LLM, facilitating efficient prompting. We then instruct the LLM to generate an *extractive* summary of the input text by selecting the most crucial and informative moments from the video. Additionally, we instruct the LLM to preserve

the original wording of the selected sentences in the summary, along with their corresponding timestamps. This ensures that the extracted textual summary can be seamlessly matched with the respective video segments. The prompt template employed in our work is illustrated in Fig. 2. We predominantly use GPT-3.5-16K [27] for large-scale dataset curation. Moreover, we conduct experimental analysis using GPT-4 [27] and Llama 2-13B [38] as summarizers.

Pseudo-Ground Truth Video Summary. We obtain the video segment corresponding to each sentence in the text summary using the *start* and *end* timestamp of the sentence in the transcript. To ensure the accurate selection of video segments that correspond to the text in the summary, thus mitigating any potential timestamp misalignments from the transcription model, we employ a CLIP embedding-based nearest neighborhood search for nearby frames within each summary video segment. Subsequently, the resulting video segments are temporally aggregated to construct a pseudo-ground truth (pGT) summary. Following the pipeline depicted in Fig. 2, we create **LfVS-P**, a large-scale dataset containing 250K videos and their associated pseudo-ground truth summaries, for video summarization pretraining. In Table 1, we compare our dataset with existing video summarization datasets. The proposed pretraining dataset stands out for its notable scale and diversity across a wide range of tasks. The longer average video duration in our dataset also facilitates robust training for summarizing videos of varying lengths.

LfVS-T Benchmark. In addition to introducing a large-scale video summarization pretraining dataset, our work establishes a new benchmark, named Long-form Video Summarization Testing (**LfVS-T**), for evaluating models. LfVS-T consists of 1200 videos, each accompanied by manually annotated, high-quality ground truth summaries from professional human annotators. The dataset is sourced from publicly available YouTube content, featuring both narrated and non-narrated videos. The video durations range from 8 to 33 minutes, covering a wide spectrum of 392 distinct categories. The size and diversity of LfVS-T (see Table 1) make it a valuable benchmark for video summarization models, facilitating further research in the field.

4. Methodology

Problem Formulation. Video summarization can be purely formulated as a video-to-video problem. However, our experimental observations have indicated a significant performance enhancement when language signal is incorporated. Therefore, we approach the task as a multi-modal problem, where we take into account both the video content and the text corpus obtained from audio transcription to generate a summary video. Let V denote a video represented as a sequence of frames uniformly sampled every t seconds, *i.e.* $V = \{X_1, X_2, \ldots, X_n\}$, where X_n denotes a frame at time step t_n and T denote the text data associated with the video represented as a sequence of sentences, *i.e.* $T = \{S_1, S_2, \ldots, S_k\}$, where S_k denotes the k^{th} sentence. Given $\{V, T\}$ as input, our model outputs a summary video $v = \{Y_1, \ldots, Y_m\}$, where $v \subset V$ and $m \ll n$. We train our network by optimizing the predicted summary v with respect to the pseudo-ground truth summary video v_{pGT}. Our approach is designed to be flexible to effectively summarize videos, whether they include speech (text) or not, at inference time (refer to Sec. 4.1).

Table 1. Comparison with different video summarization datasets.

Dataset	# of Videos	# of Tasks	Avg. Dur. (min)	Annotation
TVSum [36]	50	10	4.2	Manual
SumMe [7]	25	25	2.4	Manual
TL:DW? [26]	12.1K	185	3.1	Automatic
LfVS-P (Ours)	250K	6.7K	13.3	Automatic
LfVS-T (Ours)	1.2K	392	12.2	Manual

4.1. Video Summarization Network

We design a Transformer-based [39] encoder-decoder network for video summarization. Fig. 3 shows the overview of the proposed model. Our approach consists of four key components: long video encoding, long text encoding, cross-modal attention, and summary video decoding, which are detailed as follows.

Long Video Encoding. Learning directly from the pixel-space of long videos in an end-to-end manner is often computationally infeasible. Therefore, we opt for using state-of-the-art visual encoders for base feature extraction. Given a long-form video represented as a sequence of frames (sampled every t seconds), we use a pretrained CLIP [31] encoder to obtain a visual embedding for each video frame (see Eq. (1)). This step is equivalent to visual tokenization, wherein we transform an input video into a sequence of feature representations, *i.e.* $\{x_1, x_2, \ldots, x_n\}$. As shown in Fig. 3, we augment the visual tokens with special start-of-sequence (`SOS`) and end-of-sequence (`EOS`) tokens to mark the beginning and end of the input sequence, respectively. Next, we feed the resulting sequence into a positional encoding layer [39] to embed information regarding the relative positions of each token in the sequence.

$$\{x_1, x_2, \ldots, x_n\} = \textbf{CLIP}\big(\{X_1, X_2, \ldots, X_n\}\big) \qquad (1)$$

After this stage, we pass the resulting sequence to a video encoder (referred to as **V-Encoder** in Eq. (2)), which consists of a stack of transformer encoder layers [39]. The purpose of the video encoder is to perform temporal reasoning over the input video sequence. In this process, each video moment within the sequence, represented by a visual token, interacts with and attends to every other video moment via a

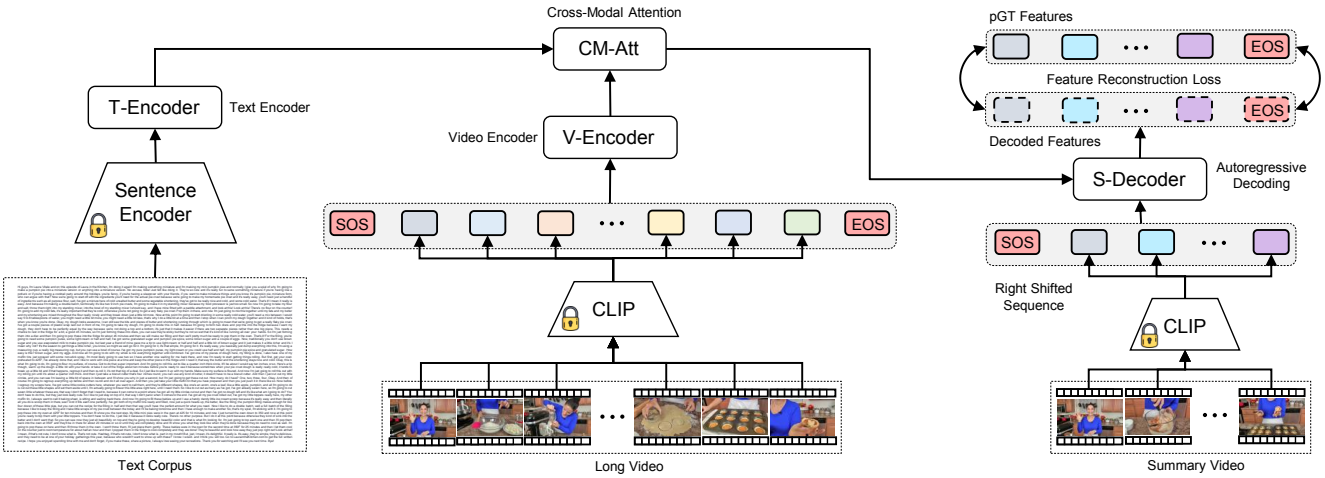

Figure 3. **Video Summarization Network.** We use a pretrained CLIP [31] model to represent an input video as a sequence of visual tokens. Similarly, we use a pretrained sentence encoder [18] to encode the long text corpus. In the absence of associated text, we utilize a special MASK token as the text input. We then use a stack of transformer encoders to contextualize the visual and textual features. Next, we incorporate multi-modal cues from the contextualized features via cross-modal attention. Finally, a summary decoder takes the multi-modal features as input and autoregressively decodes the visual representation of the segments that will compose a video summary.

self-attention mechanism. Consequently, the video encoder outputs a sequence of contextualized visual representations, *i.e.* $\{\hat{x}_0, \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{n+1}\}$.

$$\{\hat{x}_i\}_{i=0}^{n+1} = \textbf{V-Encoder}\big(\{\texttt{SOS}, x_1, \ldots, x_n, \texttt{EOS}\}\big) \quad (2)$$

Long Text Encoding. Given a transcribed text associated with an input video, we use a pretrained language model to obtain an encoded representation of the raw text. Considering the text corpus in long videos, where the number of tokens often exceeds the context size of most token-based large language models [4, 18], we employ a state-of-the-art sentence-based language model, SRoBERTa [33], for convenience (Eq. (3)).

$$\{s_1, s_2, \ldots, s_k\} = \textbf{SRoBERTa}\big(\{S_1, S_2, \ldots, S_k\}\big) \quad (3)$$

where $\{s_1, s_2, \ldots, s_k\}$ denote a sequence of extracted sentence embeddings. To further facilitate text-based contextual learning for video summarization, we pass the extracted sentence embeddings through a text encoder (denoted as **T-Encoder** in Eq. (4)), which comprises a stack of transformer encoder layers [39].

$$\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k\} = \textbf{T-Encoder}\big(\{s_1, s_2, \ldots, s_k\}\big) \quad (4)$$

where $\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k\}$ represent the encoded textual representations derived from the text encoder. We aim to design a video summarization framework capable of handling videos with or without corresponding text. To achieve this, we train our model by randomly masking the input text with a special MASK token, using a masking ratio ranging from 0 to 100 percent. This approach allows the network to learn to rely solely on video input when text input is unavailable. At inference, if the input video does not have a corresponding text, we simply use the MASK token as a text input.

Cross-Modal Attention. To capture inter-modal relationships between video and text inputs, thereby incorporating multi-modal cues for video summarization, we use a cross-modal attention module. Specifically, we adopt the multi-head attention mechanism proposed in [39] with minor modifications, where we use the encoded visual features as query (Q) vector and the encoded text features as key (K) and value (V) vectors. Let $\hat{x}$ and $\hat{s}$ denote the outputs of the video encoder (Eq. (2)) and text encoder (Eq. (4)), respectively. The attention mechanism within each head is then defined as follows:

$$\text{head} = \text{Att.}(\hat{x}W^Q, \hat{s}W^K, \hat{s}W^V) \quad (5)$$

$$\text{Att.}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (6)$$

where W^Q, W^K, and W^V denote learned parameter matrices and d_k is the size of K. The cross-modal attention module (denoted as **CM-Att** in Eq. (7)) produces text-conditioned visual features. These features are subsequently utilized as context in a decoder network to generate the video summary.

$$\{\hat{x}_i^{\hat{s}}\}_{i=0}^{n+1} = \textbf{CM-Att}\big(\{\hat{x}_i\}_{i=0}^{n+1}, \{\hat{s}_j\}_{j=1}^{k}\big) \quad (7)$$

Summary Video Decoding. The summary decoder takes the multi-modal features from the cross-modal attention module as input and generates the visual embeddings of the segments that will compose a video summary in an autoregressive fashion. It comprises a series of transformer decoder layers [39]. Similar to next-word prediction in NLP, we implement a next-summary moment prediction scheme in our model. During the training phase, to decode the feature representation of a summary moment at time step t

(*i.e.* $\hat{y}_t$), the summary decoder takes the output of the cross-modal attention module as context and the target pGT summary video sequence up to time step $t-1$ as input as shown in Eq. (9). This design choice accounts for previously selected moments in the summary when choosing the next summary moment from the video. In testing, the summary decoder initiates with the context from the cross-modal attention and the SOS token. It proceeds to generate the feature representation of the summary video sequence in an autoregressive manner, using the previously generated sequence as input, until the EOS token is decoded.

$$\{y_1, y_2, \ldots, y_m\} = \textbf{CLIP}\big(\{Y_1, Y_2, \ldots, Y_m\}\big) \quad (8)$$

$$\hat{y}_t = \textbf{S-Decoder}\big(\{\hat{x}_i^{\hat{s}}\}_{i=0}^{n+1}, \{\texttt{SOS}, y_1, \ldots, y_{t-1}\}\big) \quad (9)$$

Training and Inference. We train our network by optimizing the feature reconstruction loss between the predicted video summary $(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m)$ and the pseudo-ground truth summary $(y_1, y_2, \ldots, y_m)$ as follows,

$$\mathcal{L} = \sum_{i=1}^{m+1} \big|\hat{y}_i - y_i\big|^2 \quad (10)$$

where y_{m+1} denotes the EOS token. In inference, we utilize nearest neighbor retrieval to match the decoded summary video representations with the CLIP embeddings of the input video sequence. This process selects relevant video moments, which are then temporally aggregated to form the video summary.

5. Experiment

Implementation Details. We sample videos at a rate of 1 frame per second (1 fps) to represent input videos and pseudo-ground truth summaries as a sequence of frames. We use CLIP-ViT-L/14 [31] for visual tokenization and SRoBERTa-NLI-large [33] for sentence embedding extraction. Our architecture consists of a video encoder with 6 transformer encoder layers [39], a text encoder with 3 transformer encoder layers, a single cross-modal attention layer, and a summary video decoder with 6 transformer decoder layers [39]. Each encoder and decoder layer has a hidden dimension of 1024, 8 attention heads, and a feed-forward dimension of 2048. We train our model using the AdamW optimizer [19] with a cosine learning rate annealing strategy [37], starting from an initial learning rate of $3e-4$. The training utilizes a mini-batch size of 64 and runs for 100 epochs on 4 NVIDIA A6000 GPUs.

Evaluation Datasets and Metrics. We evaluate our approach, along with state-of-the-art video summarization models [8, 10, 25, 26], on established benchmarks such as TVSum [36] and SumMe [7], in addition to the newly introduced LfVS-T benchmark. To measure video summarization performance, we follow established practices [8, 10, 26] and utilize three different metrics: F1-score, Kendall's τ [13], and Spearman's ρ [50] metrics.

Table 2. **Experimental comparison with SoTA approaches**. We train each model on the LfVS-P dataset and evaluate their performance using the proposed LfVS-T benchmark.

Method	F1 Score	τ [13] Metric	ρ [50] Metric
CLIP-It [25]	62.87	0.129	0.225
TL:DW? [26]	<u>66.25</u>	0.138	0.233
iPTNet [10]	65.80	0.140	0.237
A2Summ [8]	66.04	<u>0.143</u>	<u>0.246</u>
Ours	**68.11**	**0.158**	**0.277**

5.1. Experimental Results

We evaluate our approach against several state-of-the-art video summarization models, including CLIP-It [25], TL:DW? [26], iPTNet [10], A2Summ [8]. For a fair comparison, we evaluate all methods using the same experimental settings, adhering to their official implementations [1]. To adapt previous models [8, 10, 25, 26] formulated for predicting the importance score of each frame (segment) in a video sequence to our experiment setup, we generate ground truth importance scores as follows: we compute the cosine similarity between the CLIP embedding of each video frame X_i (*i.e.* x_i) in the input sequence and the CLIP embedding of every frame in the summary sequence (*i.e.* $\{y_1, y_2, \ldots, y_m\}$), and assign the maximum value as the importance score z_i for X_i as shown in Eq. (11).

$$\{z_i\}_{i=1}^{n} = \max_j s_{i,j}, \quad \text{where } s_{i,j} = \frac{x_i \cdot y_j}{\|x_i\|\|y_j\|}, \quad (11)$$

A high value of z_i indicates that a video frame X_i is considered important and included in the summary, while a low value suggests that the video moment is dissimilar to any frames in the summary, implying low importance.

Comparison with State-of-the-Art. We train our approach and existing video summarization methods on the LfVS-P dataset and evaluate their performances on the proposed LfVS-T benchmark. The results are summarized in Table 2. As evident from the table, video summarization approaches that integrate text information, such as TL:DW? [26], A2Summ [8], and ours, generally outperform video-only methods like iPTNet [10]. This is intuitive, as the addition of text information provides extra context, confirming the benefit of framing video summarization as a multi-modal problem.

As shown in Table 2, our approach achieves a notably better performance compared to state-of-the-art models across all metrics. For instance, on the F1-score metric, our model outperforms TL:DW? and A2Summ by 2.8%

[1] We reimplement CLIP-It [25] and iPTNet [10] as their official code is not publicly available.

Table 3. **Results on SumMe and TVSum datasets**. We compare our work and previous methods using the canonical train/test split of SumMe and TVSum datasets. We also conduct cross-dataset generalization experiments by training our model on the LfVS-P dataset and evaluating it on the two datasets.

	SumMe [7]			TVSum [36]		
Method	F1 Score	τ [13]	ρ [50]	F1 Score	τ [13]	ρ [50]
Human [28]	54.00	0.205	0.213	78.00	0.177	0.204
CLIP-It [25]	54.47	0.109	0.120	**66.49**	0.116	0.159
TL:DW? [26]	56.46	0.111	0.128	65.84	0.143	0.167
iPTNet [10]	56.61	0.114	0.131	66.16	0.148	0.174
A2Summ [8]	**57.09**	0.121	0.143	66.10	0.150	0.178
Ours	56.94	**0.130**	**0.152**	66.04	**0.155**	**0.186**
Cross-dataset						
Ours (zero-shot)	56.72	0.125	0.148	65.76	0.151	0.182
Ours (fine-tuned)	**60.42**	**0.147**	**0.171**	**72.38**	**0.169**	**0.203**

and 3.1%, respectively. While previous works predict discrete importance scores for each frame in the input video sequence, our model is designed to decode continuous feature representations of the summary moments. This approach offers benefits in mitigating the inherent class imbalance in video summarization tasks as it allows flexibility in determining how to represent and generate the summary rather than being confined to discrete classes (refer to Sec. 5.3). More importantly, unlike existing approaches that predict importance scores for all input frames in parallel, our autoregressive model enables conditional generation. Each summary moment is generated based on both the input context and previously generated summary moments, a crucial aspect in video summarization where context is essential for generating subsequent summary moments, contributing to the superior results observed in Table 2. Please refer to the supplementary for further qualitative analysis.

Results on SumMe and TVSum Datasets. In Table 3, we evaluate our approach and state-of-the-art methods on SumMe [7] and TVSum [36] benchmarks. Following previous works [8, 10, 25], we evaluate all methods under canonical train-test splits, conducting experiments five times and reporting the averaged results. As shown in Table 3, our approach demonstrates highly competitive, if not superior, performance on both SumMe and TV-Sum datasets. In particular, our approach notably outperforms previous works on Kendall's τ [13], and Spearman's ρ [50] metrics, which gauge the correlation between predicted and ground truth video summary sequences. These results highlight the benefits of the proposed decoding approach which enables our model to capture sequential dependencies between summary moments when generating a video summary.

We also conduct cross-dataset generalization experiments, where we train our model on the LfVS-P dataset and evaluate it on SumMe [7] and TV-Sum [36] test sets in both zero-shot and fine-tuned settings. As can be seen from Table 3, our model, pretrained on pseudo-ground truth summaries, achieves a competitive zero-shot performance on both datasets despite the domain gap. Fine-tuning our pretrained model on the training splits of SumMe and TV-Sum leads to a substantial improvement, establishing a new state-of-the-art in video summarization on both datasets. For instance, on the F1-score metric, our fine-tuned model surpasses the performance of the model trained from scratch by 6.1% and 9.1% on the SumMe and TVSum datasets, respectively. This underscores the benefits of our proposed dataset curation framework, which is designed to achieve robust video summarization through large-scale pretraining.

Table 4. **Ablation studies** on LfVS-T benchmark.

Method	F1 Score	τ [13] Metric	ρ [50] Metric
Video (w/o Text Input)	66.59	0.152	0.268
w/o Text & Video Encoder	62.77	0.133	0.231
w/o Video Encoder	63.54	0.141	0.240
w/o Text Encoder	67.49	0.154	0.272
w/o Cross-Attention	67.72	0.155	0.274
Full Model	**68.11**	**0.158**	**0.277**

5.2. Ablation Studies

In Table 4, we conduct ablation experiments on various network components in our video summarization network. Each model variant is trained on the LfVS-P dataset, and its performance is evaluated on the LfVS-T benchmark.

Text Input. To investigate the significance of incorporating text for video summarization training, we input the contextualized output from the video encoder directly into the summary decoder, omitting the use of a text encoder and cross-modal attention (refer to Fig. 3). This method frames video summarization as a video-to-video problem, focusing solely on visual information. The results are summarized in Table 4. As evident from the table, our video (without text input) baseline performs reasonably well. However, incorporating text input during pretraining to guide video summary generation results in notable improvements. For instance, on the F1 score metric, the baseline trained with text input outperforms the text-less baseline by 2.3%. The results highlight the benefit of approaching video summarization training as a multi-modal problem, rather than adhering to a pure video-to-video formulation.

Video Encoder. Here, we explore the importance of the video encoder in our framework by directly inputting the sequence visual tokens extracted from a pretrained CLIP [31] model into the cross-modal attention module. As can be inferred from Table 4, a baseline without a video encoder significantly underperforms compared to the full model. A similar pattern is observed with a baseline lacking both text and video encoders. This is mainly because the input visual tokens are extracted independently, and feeding them directly to the summary decoder without learning their contextual dependencies via the video encoder provides a much less meaningful context to the summary decoder. Conse-

quently, this leads to a subpar video summarization performance, as shown in Table 4.

Text Encoder. To evaluate the effectiveness of text-based contextual learning for video summarization through the text encoder, we train our network by directly inputting the sentence embeddings extracted from a pre-trained SRoBERTa [33] model into the cross-modal attention network. While the baseline performs reasonably well, as shown in Table 4, it is evident that a network trained with a text encoder achieves superior performance. This result aligns with our intuition that the text encoder enables the network to learn additional context from the text input for video summarization.

Cross-Modal Attention. Here, we analyze the benefit of incorporating text and video cues via a cross-modal attention module for decoding video summaries. To achieve this, we concatenate the output of the video and text encoders and input it into the summary decoder. It can be inferred from Table 4 that concatenating contextualized text and video features yields competitive performance. This is expected as the decoder attends the different positions in the context sequence when generating each summary moment. However, explicitly performing cross-attention between the video and text features before feeding them to the decoder improves performance.

5.3. Experimental Analyses

Problem Formulation. Our approach frames video summarization as an autoregressive problem, sequentially decoding continuous representations for the summary video. We explore the benefits of this formulation by contrasting it with a classification-based baseline. In the baseline, we substitute the summary decoder in Fig. 3 with a classification layer that categorizes each moment in the input video sequence as a summary or not. This involves feeding the output of the cross-modal attention module into a linear layer and training the network using a cross-entropy loss. The corresponding results are presented in Table 5. The classification-based baseline, as seen in the table, significantly underperforms compared to its autoregressive counterpart. Similar to the frame importance score prediction baselines discussed in Sec. 5.1, the classification model predicts summary and non-summary moments concurrently, neglecting the sequential dependencies of summary moments. This accounts for its lower performance compared to the autoregressive model in Table 5.

Dataset Scale. We investigate the impact of scaling video summarization pretraining by training our network with different amounts of video-pseudo-ground truth summary pairs. Our experiments include using 25K and 125K training samples, accounting for 10% and 50%, respectively, of LfVS-P. As evident from Table 5, model performance expectedly increases proportionally to the size of the pretraining data. The automatic dataset curation pipeline can introduce noise, affecting the robustness of a model when trained on a small-scale dataset. On the other hand, training on a large-scale dataset provides exposure to diverse samples, contributing to a more robust model, as reflected in the results in Table 5.

Table 5. **Experimental analyses** on LfVS-T benchmark.

Method	F1 Score	τ [13] Metric	ρ [50] Metric
Problem formulation			
Classification	63.31	0.132	0.229
Autoregressive	**68.11**	**0.158**	**0.277**
Dataset Scale			
10%	53.44	0.101	0.169
50%	64.58	0.145	0.248
100%	**68.11**	**0.158**	**0.277**
LLM (50K samples)			
Llama-2-13B	44.89	0.088	0.137
GPT-3.5-16K	53.44	0.101	0.169
GPT-4	**55.96**	**0.123**	**0.181**

LLM (50K Samples). The prompt-tuned extractive text summarization using LLMs, illustrated in Fig. 1 and Fig. 2, is a crucial step in our pseudo-ground truth video summary curation process. Here, we examine how employing different LLMs as oracle summarizers for generating pretraining data influences video summarization performance. We utilize three state-of-the-art LLMs, Llama 2-13B [38], GPT-3.5-16K [27], and GPT-4 [27], to generate 50K training samples (with the same set of input videos for each case) and subsequently train our model. The results are shown in Table 5. As can be inferred from the table, a model trained on a dataset obtained using GPT-4 as a summarizer achieves the best performance. This is expected, as GPT-4 has demonstrated superior capabilities in comprehending and summarizing long text corpora [3], resulting in the generation of high-quality pseudo-ground truth summaries. In contrast, a model trained on a dataset generated using Llama-2-13B [38] exhibits subpar performance. Our experimental observations indicate that the Llama-2-13B model struggles to precisely follow prompt instructions, leading to the generation of low-quality summaries.

6. Conclusion

This work introduces an automatic, scalable mechanism using long-form videos and LLMs to create the LfVS-P dataset for large-scale video summarization pretraining. We also propose an autoregressive video summarization model that effectively addresses previous limitations. Additionally, we present the LfVS-T benchmark, comprising 1,200 long videos with human-annotated high-quality summaries. Our extensive comparisons with previous methods demonstrate that our work establishes a new state-of-the-art in video summarization across several benchmarks.

References

[1] Suad Alhojely and Jugal Kalita. Recent progress on text summarization. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1503–1509. IEEE, 2020. 2

[2] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747*, 2023. 1, 3

[3] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 8

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[5] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1600–1607. IEEE, 2012. 1, 2

[6] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, and Bahareh Gholamzadeh. A comprehensive survey on text summarization systems. In *2009 2nd International Conference on Computer Science and its Applications*, pages 1–6. IEEE, 2009. 2

[7] 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. 1, 2, 4, 6, 7

[8] 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, 3, 6, 7

[9] 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. 1, 2

[10] 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, 6, 7

[11] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on artificial intelligence*, pages 8537–8544, 2019. 2

[12] Atsushi Kanehira, Luc Van Gool, Yoshitaka Ushiku, and Tatsuya Harada. Aware video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7435–7444, 2018. 3

[13] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. 6, 7, 8

[14] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1346–1353. IEEE, 2012. 1, 2

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 2

[16] Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. Videoxum: Cross-modal visual and textural summarization of videos. *IEEE Transactions on Multimedia*, 2023. 3

[17] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019. 2

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 5

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[20] Shiyang Lu, Zhiyong Wang, Tao Mei, Genliang Guan, and David Dagan Feng. A bag-of-importance model with locality-constrained coding based feature learning for video summarization. *IEEE Transactions on Multimedia*, 16(6): 1497–1509, 2014. 1, 2

[21] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017. 2

[22] James Manyika. An overview of bard: an early experiment with generative ai, 2023. 1, 2, 3

[23] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 1, 3

[24] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004. 2

[25] 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, 3, 6, 7

[26] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl; dw? summarizing instructional videos with task relevance and cross-modal saliency. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 1, 2, 3, 4, 6, 7

[27] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023. 1, 2, 3, 4, 8

[28] 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. 7

[29] Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. Llm is like a box of chocolates: the non-determinism of chatgpt in code generation. *arXiv preprint arXiv:2308.02828*, 2023. 2

[30] Jielin Qiu, Jiacheng Zhu, William Han, Aditesh Kumar, Karthik Mittal, Claire Jin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Bo Li, et al. Multisum: A dataset for multimodal summarization and thumbnail generation of videos. *arXiv preprint arXiv:2306.04216*, 2023. 3

[31] 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. 3, 4, 5, 6, 7

[32] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 1, 3

[33] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 5, 6, 8

[34] Aidean Sharghi, Boqing Gong, and Mubarak Shah. Query-focused extractive video summarization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 3–19. Springer, 2016. 3

[35] Aidean Sharghi, Jacob S Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4788–4797, 2017. 1, 2, 3

[36] 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. 1, 2, 4, 6, 7

[37] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 6

[38] 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. 1, 2, 3, 4, 8

[39] 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. 2, 4, 5, 6

[40] Yiming Wang, Zhuosheng Zhang, and Rui Wang. Element-aware summarization with large language models: Expert-aligned evaluation and chain-of-thought method. *arXiv preprint arXiv:2305.13412*, 2023. 2

[41] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020. 2

[42] 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

[43] Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 383–399, 2018. 2

[44] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*, 2023. 2

[45] Bin Zhao and Eric P Xing. Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2513–2520, 2014. 1, 2

[46] 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

[47] 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. 1, 2

[48] Li Zhong and Zilong Wang. A study on robustness and reliability of large language model code generation. *arXiv preprint arXiv:2308.10335*, 2023. 2

[49] 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

[50] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999. 6, 7, 8