

VideoXum: Cross-modal Visual and Textural Summarization of Videos

Jingyang Lin^{1*} Hang Hua^{1*} Ming Chen² Yikang Li² Jenhao Hsiao²
Chiuman Ho² Jiebo Luo¹

¹University of Rochester

²OPPO US Research Center

jlin81@ur.rochester.edu {hhua2, jluo}@cs.rochester.edu

{cmelf0819, lyk010632, mhsiao.pro, chiuman100}@gmail.com

videoxum.github.io

Abstract

*Video summarization aims to distill the most important information from a source video to produce either an abridged clip or a textual narrative. Traditionally, different methods have been proposed depending on whether the output is a video or text, thus ignoring the correlation between the two semantically related tasks of visual summarization and textual summarization. We propose a new joint video and text summarization task. The goal is to generate both a shortened video clip along with the corresponding textual summary from a long video, collectively referred to as a cross-modal summary. The generated shortened video clip and text narratives should be semantically well aligned. To this end, we first build a large-scale human-annotated dataset – **VideoXum** (X refers to different modalities). The dataset is reannotated based on ActivityNet. After we filter out the videos that do not meet the length requirements, 14,001 long videos remain in our new dataset. Each video in our reannotated dataset has human-annotated video summaries and the corresponding narrative summaries. We then design a novel end-to-end model – VTSUM-BLIP to address the challenges of our proposed task. Moreover, we propose a new metric called VT-CLIPScore to help evaluate the semantic consistency of cross-modality summary. The proposed model achieves promising performance on this new task and establishes a benchmark for future research.*

1. Introduction

Traditional video summarization aims to distill the most important information from a source video to produce an abridged version for particular users and tasks. Typical approaches to video summarization extract essential clips or frames from a given long video [2, 48, 87, 88]. Alternatively,

the principal video content can also be summarized in natural language, e.g., video captioning [11, 18, 59]. However, previous works treat either video or textual summarization as separate tasks and thus ignore bridging the two modalities of summarization tasks together. There is an earlier attempt [8] to simultaneously generate video and text summaries from long videos. However, the generated video and text summaries in this work are not guaranteed to be semantically aligned because the two tasks were treated as separate, and there were no paired video and text summarization data for training or testing.

In this study, we first propose **VideoXum**, an enriched large-scale dataset for cross-modal video summarization. The dataset is built on ActivityNet Captions [29], a large-scale public video captioning benchmark. We hire workers to annotate ten shortened video summaries for each long source video according to the corresponding captions. VideoXum contains 14K long videos with 140K pairs of aligned video and text summaries. Next, our goal is to extend the traditional single-modal video summarization task to a cross-modal video summarization task, referred to as V2X-SUM, to meet the demands of broader application scenarios (e.g., movie trailer generation and narrative generation). According to the target modality of the generated summaries, we categorize our proposed V2X-SUM task into three subtasks:

Video-to-Video Summarization (V2V-SUM). This task requires models to identify the most important segments from the source video and generate an abridged version of the source video.

Video-to-Text Summarization (V2T-SUM). In this task, models need to summarize the main content of the source video and generate a short text description.

Video-to-Video&Text Summarization (V2VT-SUM). This task requires models to summarize a short video and the corresponding narrative from a source video simultaneously. Moreover, the semantics of these two modalities of summaries should be well aligned.

*Equal contribution

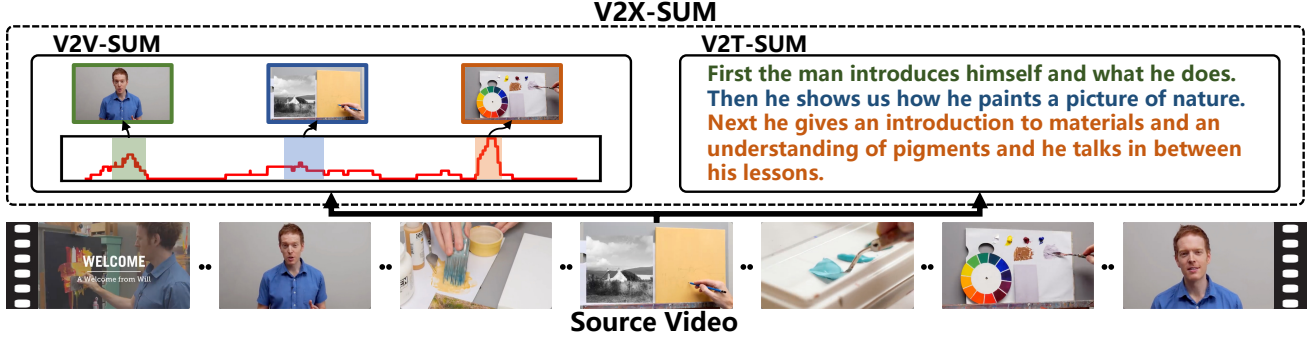


Figure 1: Illustration of our V2X-SUM task. A long source video (*bottom*) can be summarized into a shortened video and a text narrative (*top*). The video and text summaries should be semantically aligned.

There are three potential challenges for the V2X-SUM task from the perspective of language model learning: First, simultaneously generating video and text summaries for a vision and language model could be quite challenging due to the time and GPU memory efficiency requirement for video encoding and the stability requirement for language model fine-tuning; Second, from the perspective of model design and optimization, it is nontrivial for the combined two tasks to learn and benefit from each other; Third, it is hard to guarantee the generated different modalities of the cross-modal summary are semantically aligned.

We propose VTSUM-BLIP, a novel end-to-end cross-modal video summarization model, to tackle the above challenges. To leverage the strong capability of vision understanding and language modeling of pretrained language models, we employ BLIP [33] as our backbone. Inspired by HERO [34], we design an efficient hierarchical video encoding strategy with a frozen encoder and a temporal modeling module to encode long videos. We also design different task-specific decoders for video and text summarization. The modularized design enables us to perform more complex downstream tasks without changing the structure of the pretrained model. Our proposed framework achieves promising performance on VideoXum, as well as other existing single-modal video summarization datasets (e.g., TV-Sum [59], SumMe [18], and ActivityNet Captions [29]).

Several multimodal-based video summarization works [26, 43, 48, 52, 88] exploit video captions to guide models to predict video summaries. Such methods usually design a pipeline that generates video captions using pretrained language models and then uses the generated caption as a prompt to improve the models’ performance. However, these methods may suffer from bias accumulation issues since the hallucination [25] of pretrained language models may lead to the misalignment of semantics between the generated captions and video. In contrast, our proposed new dataset enables joint training of the video and text summarization tasks using an end-to-end model. The experiments show that our multitask training framework can improve the

model’s performance on both visual and textual summarization of videos.

Furthermore, we design a new metric – VT-CLIPScore for evaluating the semantic consistency of cross-modal summaries. The empirical results show the consistency of the proposed new metric with human evaluation.

Our main contributions can be summarized as follows:

- We propose **VideoXum**, an enriched large-scale dataset, to bridge the modality gap between the video and text summarization. The dataset contains 14,001 long videos with corresponding human-annotated video and text summaries. We conduct comprehensive experimental analyses to verify the rationality of our proposed new dataset.
- We propose a novel end-to-end video and text summarization model – VTSUM-BLIP to address the challenges of our proposed task of cross-modal summarization. The model achieves promising results on VideoXum and the new state of the art on several existing single-modal video summarization datasets.
- We propose an evaluation metric VT-CLIPScore to evaluate cross-modal semantic consistency. The empirical results show the high consistency of our proposed metric with human evaluation.

2. Related Work

2.1. Video Summarization

Video summarization datasets, e.g., SumMe [18], TV-Sum [59], and VSUMM [11] have enabled the development of state-of-the-art video summarization models. Among these models, vsLSTM [77] first attempted to learn frame importance by modeling the temporal dependency among frames using LSTM [17] units. The model can be combined with a determinantal point process (DPP) to improve the diversity of generated video summary. Following vsLSTM, several other approaches were proposed to model the temporal dependency, e.g., H-RNN [80], Hsa-RNN[81], TTH-RNN [82], DASP [24]. Another solution models the spa-

Table 1: Comparison with existing single-modal video-to-video summarization and video-to-text summarization datasets.

Dataset	Domain	# Videos	Avg Ratio (%) VideoSum	Avg Len TextSum	Supported Task		
					V2V-Sum	V2V-Sum	V2VT-Sum
YouCook [85]	cooking	88	-	15.9	✗	✓	✗
MSR-VTT [71]	open	10K	-	18.6	✗	✓	✗
ActivityNet [29]	open	20K	-	40	✗	✓	✗
SumMe [18]	3 categories	25	15% <	-	✓	✗	✗
TVSum [59]	10 categories	50	15% <	-	✓	✗	✗
VideoXum (ours)	open	14K	13.6%	49.9	✓	✓	✓

tiotemporal structure of the video to learn frame importance, such as MerryGoRoundNet [30], and CRSum [75]. Adversarial learning-based methods [15, 79] can also perform well in video summarization.

2.2. Video Captioning

Video Captioning aims to automatically generate short descriptions for a video by understanding the action and event in a video, which can help retrieve videos efficiently through text queries. Existing benchmarks (e.g., YouCook2 [84], DiDeMo [19], MSR-VTT [71] ActivityNet Captions [29], VATEX [68], and MSVD[9]) have helped to promote the ability of language models to generate reasonable captions for video. Benefiting from these large-scale datasets, many novel approaches are proposed and achieve state-of-the-art performance. MDVC [23] utilizes audio and speech modalities to improve dense video captioning.

2.3. Multimodal Pretraining

Large language models (LLMs) [7, 12, 32, 50] have revolutionized NLP research in recent years. Following the large-scale pretraining models in the field of NLP, numerous works [28, 67, 72, 78] on exploring the combination of vision and language (VL) pretraining have achieved great success. Since then, image-text pretraining has become a default approach to tackling VL tasks [6, 38, 51, 58]. In addition, the introduction of Vision Transformers [13] enables vision and language modalities to be jointly modeled by Transformers in a more scalable fashion [1, 65, 73, 74]. According to the encoding strategies for image and language modalities, VL models can be categorized into fusion encoder [35, 41, 60, 63], dual encoder [49], and a combination of both [5, 14, 57]. Several pretrained vision and language models have also shown strong performance on video captioning and other tasks, such as HERO [34], VideoBERT [61], and UniVL [42].

3. Dataset

In this section, we introduce the proposed VideoXum dataset. The dataset is annotated by workers, including 14,001 long videos with video and text summaries pairs.

We describe the process of dataset collection and annotation strategy. In addition, we provide several quantitative and qualitative analyses of the proposed dataset.

3.1. Dataset Curation

3.1.1 Dataset Collection

The VideoXum dataset is built based on ActivityNet Captions [29], a high-quality public video captioning benchmark. The dataset contains 20K real-life Youtube videos with diverse content, in terms of abundant topics, different photographic devices, multiple view angles, and so on. Each video in this dataset is annotated with a series of dense temporal segments, and each segment corresponds to a concrete sentence description. As described in Section 1, the well-annotated sentence narratives are natural summaries of the source videos. The content and length of videos in the ActivityNet Captions dataset largely meet our requirements for building the cross-modal summarization dataset and provide an ideal foundation for constructing our cross-modal summarization benchmark. We filter out videos shorter than 10 seconds.

3.1.2 Dataset Reannotation

In our video and text summarization task, each cross-modal video summary contains multiple summarized video spans and the corresponding sentence descriptions. For each video, we expect the total length of its video summary to be bounded to 15% of the source video. ActivityNet Captions [29] already contains video captions with temporal segments for long videos. Therefore, we concatenate the caption sentences as a text summary for the long source video. However, the annotated video spans, which cover an average of 94.6% of the source videos, are too long to be regarded as a video summary by themselves since video summaries need to be much more concise. Therefore, we reannotate the video spans and obtain an abridged version of video segments for better aligning with the sentence captions. Due to the inherently subjective nature of summarizing a long video (this conclusion is also reflected by the human performance on V2V-SUM in Table 3), it is hard to obtain perfect ground truth labels for this task. Following previous works [31, 59, 62], we employ ten different

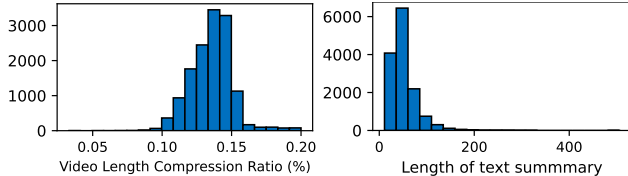


Figure 2: Distributions of video length compression ratio (*left*) and length of text summary (*right*).

workers to annotate the video summary spans for each text description. During the evaluation of video summarization, we will compare the prediction with all ten annotations and then obtain the average score for each video. We then filter the initial ActivityNet dataset using the length compression ratio of video summaries with 20% as the threshold. The video length compression ratio is calculated as $\text{Ratio}(S, V) = \frac{|S|}{|C|}$, where $|S|$ denotes the length of summary, $|C|$ denotes the length of source video. Finally, 14,001 long videos remain in our dataset.

3.1.3 Dataset Split

We split the dataset into training, validation, and test sets. The split strategy also guarantees that all three data splits preserve the same distribution of video length. In particular, the whole dataset is divided into 8,000, 2,001, and 4,000 videos in the training, validation, and test sets, respectively.

3.2. Dataset Statistics

The VideoXum dataset consists of 14,001 videos in total. The length of the videos ranges from 10 to 755 seconds, with most of them under 300 seconds. Each long video has 10 corresponding video and text summary pairs. The distributions of video length compression ratio and text summary length among the dataset are shown in Figure 2. For the video summarization task, most video summary lengths are shorter than 15% of the source video length. The average length compression ratio is 13.6%, with a median ratio of 13.7%, and a maximum ratio of 20%. For the text summarization task, each video is summarized into a narrative paragraph that describes multiple events. On average, each narrative paragraph contains 49.9 words. Figure 2 indicates that most (98%) text summaries are shorter than 128 words.

3.3. Comparison with Existing Single-modal Video Summarization Datasets

In Table 1, we compare the proposed VideoXum dataset with existing *single-modal* video-to-video and video-to-text¹ summarization datasets. The main difference between VideoXum and other existing datasets is that VideoXum contains aligned video and text summaries, while others

¹In this paper, we regard the video captioning task as video-to-text summarization task

Table 2: Comparison of F1 score on *human annotations*. F1-avg denotes the averaged F1 score across all reference summaries. F1-max represents the maximum F1 score. Symbol [#] denotes the results directly quoted from [44].

VideoXum (ours)		SumMe		TVSum	
F1-Avg	F1-Max	F1-Avg	F1-Max	F1-Avg	F1-Max
36.2	59.5	31 [#]	54 [#]	54 [#]	78 [#]

only have single-modal summaries for source videos. Compared with the existing video summarization benchmarks (e.g., SumMe [18] and TVSum [59]), the amount of data in the VideoXum dataset is significantly larger. In addition, VideoXum contains open-domain videos with more diverse scenarios than other datasets. To ensure the quality of human annotation, we evaluate the annotated data using a leave-one-out strategy [44]. Table 2 shows that our annotation quality is comparable with existing benchmarks.

4. Methodology

4.1. Problem Formulation

We formulate the problem of cross-modal video summarization as a multi-task learning problem, including V2V-SUM and V2T-SUM. Given a video $\mathcal{V} = \{v_i\}_{i=1}^T$, T is the number of frames in the video and v_i denotes the i -th frame in the temporal order. Our goal is to learn a shared video encoder $f(\cdot; \theta)$ followed by two task-specific decoders, including a video summarization decoder $g_v(\cdot; \theta_v)$ and a text summarization decoder $g_t(\cdot; \theta_t)$. In particular, the notations of θ , θ_v , and θ_t represent the learnable parameters of the shared video encoder, video summarization decoder, and text summarization decoder, respectively. We first feed the input video \mathcal{V} into the shared video encoder to produce the video features $\tilde{\mathcal{Z}}$:

$$\tilde{\mathcal{Z}} = f(\mathcal{V}, \mathcal{E}_{\text{temp}}; \theta), \quad (1)$$

where $\mathcal{E}_{\text{temp}}$ is the temporal position embedding for the video frames. Given the video features $\tilde{\mathcal{Z}}$, the model generates a video summary \mathcal{V}_{sum} and a text summary \mathcal{T}_{sum} . In particular, we can formulate visual and narrative outcomes as follows:

$$\mathcal{V}_{\text{sum}} = g_v(\tilde{\mathcal{Z}}; \theta_v), \quad (2)$$

$$\mathcal{T}_{\text{sum}} = g_t(\tilde{\mathcal{Z}}, \mathcal{T}_{\text{prompt}}; \theta_t), \quad (3)$$

where $\mathcal{T}_{\text{prompt}}$ denotes a prompt sequence.

4.2. Cross-modal Video Summarization

Our proposed VideoXum benchmark requires models with strong video understanding and language modeling capabilities. To this end, we employ the large pretrained vision-language model BLIP [33] as our backbone. The framework of our model is shown in Figure 3.

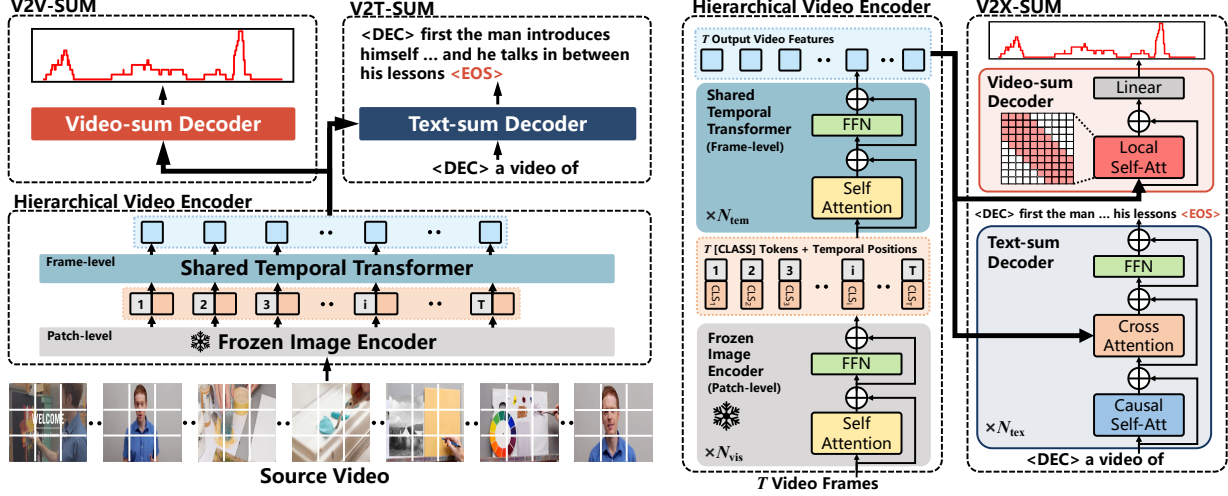


Figure 3: An overview of our VTSUM-BLIP model (left). It consists of a hierarchical video encoder (middle), video-sum decoder, and text-sum decoder (right). For V2V-SUM, the video-sum decoder employs a temporal Transformer and local self-attention module to aggregate the local context. For V2T-SUM, the text-sum decoder is a pretrained BLIP text decoder.

4.2.1 Hierarchical Video Encoder

The hierarchical video encoder $f(\cdot; \theta)$ aims to address the challenge of efficiently extracting temporal-aware visual features from a long video. Drawing the inspiration from long document summarization [53], we formulate the BLIP image encoder into a hierarchical architecture for long video encoding without changing the structure of the encoder. This enables us to efficiently obtain rich video features at the frame and image patch levels. Specifically, given a video $\mathcal{V} = \{v_i\}_{i=1}^T$ with T -frame, the frozen image encoder projects each video frame v_i into the representation space and produce T visual tokens $\mathcal{Z} = \{z_i\}_{i=1}^T$. Next, we use temporal position embedding $\mathcal{E}_{\text{temp}} = \{e_i\}_{i=1}^T$ with the shared Temporal Transformer (TT) to model the temporal information for the video sequence. In this way, we can obtain temporally-aware visual features $\tilde{\mathcal{Z}} = \{\tilde{z}_i\}_{i=1}^T$.

Frozen Image Encoder. Following the previous works [10, 78], we freeze the parameters of the pretrained BLIP encoder, which can help to improve the training time and GPU memory efficiency for encoding long videos. In detail, we first convert input images into several patches as the input tokens for the N_{vis} -layer BLIP encoder. The patch embedding is prepended with a [CLS] token in the representation space. Next, we take all output of the [CLS] tokens as the representation of the input frames. We can compress the input video at the frame level through the hierarchical encoding strategy and generate the representation \mathcal{Z} .

Shared Temporal Transformer. After obtaining the representation of the sequence of the video frame $\mathcal{Z} = \{z_i\}_{i=1}^T$, we add these temporal position embeddings $\mathcal{E}_{\text{temp}} = \{e_i\}_{i=1}^T$ to \mathcal{Z} , and feed them into the shared temporal Transformer (TT) for temporal modeling and get the temporal-aware visual features $\tilde{\mathcal{Z}} = \{\tilde{z}_i\}_{i=1}^T$ in Eq.(1):

$$\begin{aligned} z_i^{(0)} &= z_i + e_i, \\ z_i^{(l)} &= \text{TT}^{(l)}(z_1^{(l-1)}, \dots, z_T^{(l-1)}), \quad l = 1, \dots, N_{\text{tem}}, \\ \tilde{z}_i &= z_i^{(N_{\text{tem}})}, \end{aligned} \quad (4)$$

where l indicates the l -th block of the temporal Transformer, and N_{tem} denotes total block number of the temporal Transformer.

4.2.2 Video-Sum Decoder

The video-sum decoder $g_v(\cdot; \theta_v)$ contains a Context Aggregation (CA) module that captures context from neighboring frames with local self-attention. In particular, we first define a fixed-size slice window at each temporal position and then construct a binary local attention map $M^{LA} \in \{0, 1\}^{T \times T}$ with a given window size ε . For example, Figure 3 (right) presents a local attention map with a window size $\varepsilon = 7$. Next, we compute the local attention features \mathcal{A}_{loc} :

$$\mathcal{A}_{\text{loc}} = \left(\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \odot M^{LA} \right) V, \quad (5)$$

where \odot is element-wise multiplication, and queries Q , keys K , and values V are d -dimensional features generated from temporal-aware visual features $\tilde{\mathcal{Z}}$. Finally, we feed local attention-enhanced features into a linear classifier to obtain the predictions of the frame-level importance scores $\{p_i\}_{i=1}^T$. Our training objective for video summarization is an averaged binary cross-entropy loss, as following:

$$\mathcal{L}_v = -\frac{1}{T} \sum_{i=1}^T (\hat{y}_i \cdot \log(p_i) + (1 - \hat{y}_i) \cdot \log(1 - p_i)), \quad (6)$$

where $\hat{y}_i \in \{0, 1\}$ denotes whether the i -th frame is a key frame, and p_i indicates the predicted importance score of

the i -th frame. Finally, we select the top 15% of frames to attain a video-sum result \mathcal{V}_{sum} in Eq.(2) from a long video.

4.2.3 Text-Sum Decoder

The pretrained BLIP text decoder is a strong baseline for text generation. The text summarization decoder $g_t(\cdot; \theta_t)$ contains N_{tex} stacked Transformer decoder blocks with cross-attention modules. During the decoding process, the text decoder takes a prompt sequence $\mathcal{T}_{\text{prompt}}$, and the video features $\tilde{\mathcal{Z}} = \{\tilde{z}_i\}_{i=1}^T$ from the video encoder as inputs and then generate the final text summary \mathcal{T}_{sum} in Eq.(3). The training objective of text summarization is negative log-likelihood (NLL), which can be expressed in the equation as:

$$\mathcal{L}_t = - \sum_{i=1}^{N_{\text{tex}}} \log P(w_i | w_0, w_1, \dots, w_{i-1}, \tilde{\mathcal{Z}}). \quad (7)$$

where w_i denotes the i -th word in the sentence, N_{tex} is the length of output sequence.

4.3. Overall Objective

Following the multi-task learning paradigm, the overall objective of our proposed framework is calculated as the integration of video-sum loss \mathcal{L}_v and text-sum loss \mathcal{L}_t :

$$\mathcal{L} = \lambda_v \mathcal{L}_v + \lambda_t \mathcal{L}_t, \quad (8)$$

where λ_v and λ_t are the weight of different summary tasks.

5. Experiments

In this section, we first introduce the baseline models and experimental setup for the proposed VideoXum dataset. Then we introduce the evaluation metrics and human evaluation strategy. In addition, we report several baseline models' performances under different settings and present a comprehensive experimental analysis to prove the effectiveness of our proposed method.

5.1. Baseline Models

We introduce all the baseline models listed in Table 3:

Frozen-BLIP refers to inference over the test set using a frozen BLIP model without training. We take this zero-shot setting performance as a lower bound for our benchmark.

VSUM-BLIP (Base) is a baseline model to perform video-to-video summary. It consists of a frozen BLIP encoder and a learnable video-sum decoder (i.e., a task-specific head over the encoder).

TSUM-BLIP (Base) is a video-to-text summary baseline. We employ the vanilla BLIP model with a frozen encoder.

VTSUM-BLIP (Base) combines the VSUM-BLIP and TSUM-BLIP modules of the model. It is comprised of a shared frozen encoder and two task-specific decoders.

Temporal Transformer (TT) is a crucial module to achieve the hierarchical encoding for videos while incorporating temporal information into a video sequence. Specifically, we use several Transformer layers combined with

temporal positional embedding to model the temporal information.

Context Aggregation (CA) is a plug-and-play module to model the video frame representations for the V2VSum task. Compared with the baseline models, this mechanism enhances the local context information for video representations and could help reduce the redundancy of video summaries.

5.2. Experimental Setup and Implementation

Our model is implemented in PyTorch. We use the AdamW [40] optimizer with an initial learning rate of $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$ to optimize the model, and the $\beta_1 = 0.9, \beta_2 = 0.999$. The batch size is $\{16, 32, 64\}$, and weight decay is 5×10^{-2} . The learning rate follows a cosine decay schedule [39] with the minimum learning rate of 0.0. We train all baseline models for 56 epochs on 4 A100 GPUs. We sample frames from the input videos at 1 fps with a uniform sampling schema for all the experimental settings. The various lengths of videos in a mini-batch are padded to equal in length. According to the average length of text summaries in Section 3, we set the maximum generation length to 128 in the text summarization task. Following previous works [18, 59], we take the video frames with predicted scores in the top 15% as the video summary. More details are described in Appendix A.

5.3. Evaluation

Video Summary Evaluation. Following previous works [43, 44, 54] for video summarization evaluation, we adopt the F1 score, Kendall's τ [27], and Spearman's ρ [89] as our automatic evaluation metrics.

Text Summary Evaluation. To evaluate the quality of generated text summaries for video **text summary**, we adopt several metrics for video captioning evaluation [70] including: BLEU [45], METEOR [4], ROUGE-L [37], CIDEr [64].

Video-text Semantic Consistency Evaluation. Apart from independently evaluating the single-modal summaries, we also evaluate the *semantic consistency* of text and video summaries. Inspired by the previous work [49, 56, 69], we design an evaluation metric – VT-CLIPScore for evaluating the text and video semantic consistency. Specifically, we finetune the vanilla CLIP model on our dataset with contrastive learning strategies that can better fit our task. The empirical results in Table 7 show that our proposed VT-CLIPScore is sensitive enough to the semantic change of video and text. Moreover, the results in Table 6 indicate the high consistency of our proposed automatic evaluation metric with human evaluation. More details are described in Appendix A.

Table 3: The performance of the baseline models on the VideoXum dataset *test* set for three different V2XSum tasks. The F1 score, BLEU@4, METEOR, ROUGE-L, CIDEr, and VT-CLIPScore are shown in %.

Method	V2V-SUM			V2T-SUM				V2VT-SUM
	F1 score	Kendall	Spearman	BLEU@4	METEOR	ROUGE-L	CIDEr	VT-CLIPScore
Frozen-BLIP	16.1	0.008	0.011	0.0	0.4	1.4	0.0	19.5
Single-Modal Video Summarization								
VSUM-BLIP (Base)	21.7	0.131	0.207	-	-	-	-	-
+ Temporal Transformer	22.1	0.168	0.222	-	-	-	-	-
+ Context Aggregation	22.2	0.172	0.228	-	-	-	-	-
+ TT + CA	23.1	0.185	0.246	-	-	-	-	-
TSUM-BLIP (Base)	-	-	-	5.5	11.7	24.9	18.6	-
+ Temporal Transformer	-	-	-	5.6	11.8	24.9	20.9	-
Cross-Modal Video Summarization								
VTSUM-BLIP (Base)	21.7	0.131	0.207	5.5	11.7	24.9	18.6	28.4
+ Temporal Transformer	22.4	0.176	0.233	5.7	12.0	24.9	22.4	28.9
+ Context Aggregation	22.2	0.172	0.228	5.5	11.7	24.9	18.6	28.6
+ TT + CA	23.5	0.196	0.258	5.8	12.2	25.1	23.1	29.4
Human	33.8	0.305	0.336	5.2	14.7	25.7	24.2	38.0

Table 4: Comparison with state-of-the-art methods on the TVSum and SumMe datasets.

Method	TVSum		SumMe	
	Kendall	Spearman	Kendall	Spearman
dppLSTM [77]	0.042	0.055	-	-
DR-DSN [83]	0.020	0.026	-	-
Sumgraph [46]	0.094	0.138	-	-
CLIP-it [43]	0.108	0.147	-	-
Standard ranker [55]	0.176	0.230	0.011	0.014
VSUM-BLIP (Base)	0.160	0.207	0.154	0.191
+ Temporal Transformer	0.182	0.239	0.266	0.330
+ Context Aggregation	0.185	0.243	0.268	0.332
+ TT + CA	0.200	0.261	0.295	0.365

5.4. Results on VideoXum

We conduct experiments on VideoXum using different baseline models. Table 3 shows the empirical results of the models on VideoXum. By comparing VSUM-BLIP (Base), TSUM-BLIP (Base), and VTSUM-BLIP (Base) with Frozen-BLIP, BLIP models show better results after finetuning on specific tasks. In all three tasks, the model with TT can help model the video sequence better, indicating that temporal information is necessary. The CA module can enhance the local information awareness of the model, which can help improve the performance of the V2V-SUM task. The performance gains of TT on V2V-SUM are more significant than that on V2T-SUM, one of the possible reasons is that the text decoder is a well-generalized model trained on a large corpus and is insensitive to the subtle changes of the input features [3, 21, 22]. A Base BLIP model combined with TT and CA achieves the state-of-the-art in our proposed three tasks. From the overall performance, our proposed multitask framework can benefit both the video and the text summarization tasks. In addition, Table 3 also shows the human performance on VideoXum.

Table 5: Comparison with state-of-the-art methods on the ActivityNet Captions dataset.

Method	ActivityNet Captions			
	BLEU@4	METEOR	ROUGE-L	CIDEr
DENSE [29]	1.6	8.9	-	-
DVC-D-A [36]	1.7	9.3	-	-
Bi-LSTM+TempoAttn [86]	2.1	10.0	-	-
Masked Transformer [86]	2.8	11.1	-	-
Support-Set [47]	1.5	6.9	17.8	3.2
TSUM-BLIP (Base)	5.5	12.1	25.1	19.7
+ Temporal Transformer	5.7	12.1	25.2	22.2

The result is obtained on human-annotated reference summaries using a leave-one-out strategy [44], which measures the average consistency of human annotators. Although humans outperform all baseline models in most evaluation metrics on different tasks (except for BLEU@4), our proposed VTSUM-BLIP archives quite competitive results, especially on the V2T-SUM task. Figure 4 visualizes some examples of generated video and text summaries.

5.5. Experimental Analysis

5.5.1 Method Comparisons on Existing Benchmarks

TVSum and SumMe. To further evaluate the effectiveness of our proposed model, we conduct experiments on two well-known video summarization datasets TVSum and SumMe. The results in Table 4 show that the BLIP-VSUM (Base) achieves competitive results against several strong baselines. Moreover, our proposed mechanisms of Temporal Transformer and Context Aggregation can further improve video summarization performance.

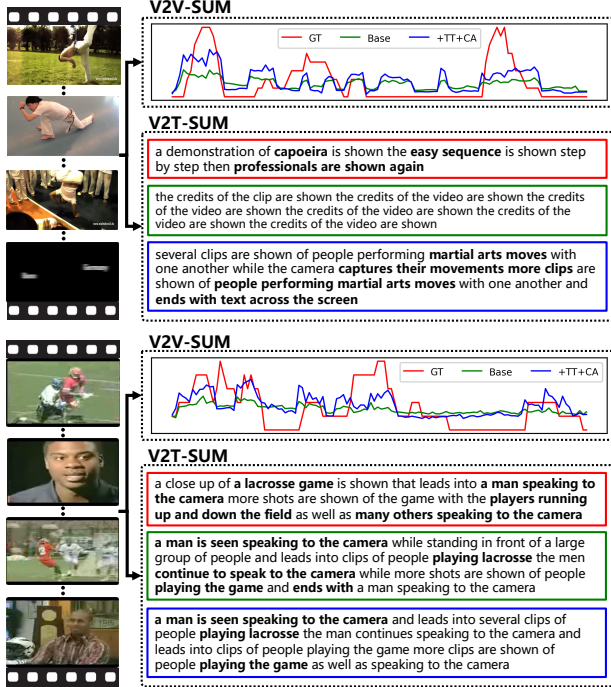


Figure 4: Two example results of the generated video and text summaries across different baseline models. Red (both line and box) indicates the results of the ground truth. Green indicates the results of the VTSUM-BLIP (Base). Blue indicates the results of VTSUM-BLIP (+TT+CA).

ActivityNet Captions. We also verify the ability of video-to-text summarization of our model on ActivityNet Captions. As we can see from Table 5, our proposed model outperforms all the strong baseline models by a large margin in multiple evaluation metrics (2.9 in BLEU@4, 1.0 in METEOR, 7.4 in ROUGE-L, and 19.0 in CIDEr).

5.5.2 Human Evaluation

We conduct the human evaluation for each task to further evaluate the quality of generated video and text summaries. Specifically, we randomly sample 50 examples from the generated summaries and then hire crowd workers to evaluate the quality of generated summaries. The crowd workers are instructed to compare the predicted video and text summaries with the source video and rate the multimodal summaries from three aspects: the quality of video summaries, the quality of text summaries, and the consistency of video and text summaries. The scores range from 1 to 5, and 5 is the best. We report the average score in Table 6. We can conclude from the table that our proposed model can generate more fluent and accurate text summaries for long videos. The proposed temporal Transformer and Context Aggregation can help generate accurate and consistent video summaries. Following [66], we compute the Kappa coefficient of different workers, and the value is $0.49 \in (0.41, 0.60)$, which means that the consistency is moderate.

Table 6: Human evaluation of the baseline models on the VideoXum dataset.

Method	V2V-SUM	V2T-SUM	Cross-Modal	
	Accuracy	Accuracy	Fluency	Consistency
VTSUM-BLIP (Base)	3.1	4.1	4.3	3.2
+ Temporal Transformer	3.5	4.2	4.2	3.2
+ Context Aggregation	3.2	4.1	4.3	3.1
+ TT + CA	3.8	4.4	4.4	3.4

Table 7: Results of cross-modality similarity under different semantic changes.

Method	Cross-Modal Similarity (Cosine Similarity)		
	Positive pairs	Negative pairs	Shuffle words
CLIP	14.5	0.3	5.4
VT-CLIPScore	38.0	0.2	28.2

5.5.3 Adapted VT-CLIPScore

Although we can apply a pretrained CLIP model without any adaptation on our dataset to evaluate the semantic consistency of the video and text summaries, the similarity score may be insensitive to the semantic change of generated cross-modal summaries. In Table 7, we compare the vanilla CLIP model and the finetuned CLIP model to measure the similarity of different video and text summarization pairs. The positive pairs refer to the paired video and text summaries. The negative pair includes unpaired video and text summaries. The Shuffle-words refer to the pairs of video summary and shuffled text summary. From the results, we can conclude that fine-tuning CLIP models on our dataset is necessary and makes the similarity scores more reflective or informative in measuring the semantic consistency of cross-modal summaries.

6. Conclusion

In this study, we first propose a new video and text summarization task along with an enriched dataset VideoXum. In this task, we jointly consider the traditional video summarization task and video-to-text summarization task. Furthermore, we propose a novel model VTSUM-BLIP to address the challenges for our proposed task. The empirical results show that our proposed framework achieves promising performance on VideoXum. In addition, we propose a new metric VT-CLIPScore to evaluate cross-modal video summarization semantic consistency, which shows high consistency with human evaluation on multiple experimental results. We plan to investigate more sophisticated models to exploit the aligned video and text summaries to improve the performance of each task. We may also employ in-context learning of language models [20] to perform cross-modal summarization.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv:2204.14198*, 2022. **3**
- [2] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *ISM*, 2021. **1**
- [3] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *ICML*, 2018. **7**
- [4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. **6**
- [5] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv:2111.02358*, 2021. **3**
- [6] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluís Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019. **3**
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. **3**
- [8] Bor-Chun Chen, Yan-Ying Chen, and Francine Chen. Video to text summary: Joint video summarization and captioning with recurrent neural networks. In *BMVC*, 2017. **1**
- [9] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. **3**
- [10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. **5**
- [11] Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern recognition letters*, 2011. **1, 2**
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. **3**
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. **3, 12**
- [14] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv:2202.10936*, 2022. **3**
- [15] Tsu-Jui Fu, Shao-Heng Tai, and Hwann-Tzong Chen. Attentive and adversarial learning for video summarization. In *WACV*, 2019. **3**
- [16] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. **12**
- [17] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 2012. **2**
- [18] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014. **1, 2, 3, 4, 6**
- [19] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. *arXiv:1809.01337*, 2018. **3**
- [20] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv:2211.09699*, 2022. **8**
- [21] Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. Noise stability regularization for improving bert fine-tuning. *arXiv:2107.04835*, 2021. **7**
- [22] Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. Fine-tuning pre-trained language models with noise stability regularization. *arXiv:2206.05658*, 2022. **7**
- [23] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *CVPR Workshops*, 2020. **3**
- [24] Zhong Ji, Fang Jiao, Yanwei Pang, and Ling Shao. Deep attentive and semantic preserving video summarization. *Neurocomputing*, 2020. **2**
- [25] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2022. **2**
- [26] 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*, 2019. **2**
- [27] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 1945. **6**
- [28] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. **3**
- [29] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. **1, 2, 3, 7, 12, 13**
- [30] Shamit Lal, Shivam Duggal, and Indu Sreedevi. Online video summarization: Predicting future to better summarize present. In *WACV*, 2019. **3**
- [31] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, 2021. **3, 12**
- [32] 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:1910.13461*, 2019. **3**

- [33] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2, 4, 12
- [34] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv:2005.00200*, 2020. 2, 3
- [35] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv:1908.03557*, 2019. 3
- [36] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018. 7
- [37] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004. 6
- [38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3
- [39] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 6
- [40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6, 12, 13
- [41] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3
- [42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv:2002.06353*, 2020. 3
- [43] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. In *NeurIPS*, 2021. 2, 6, 7
- [44] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkilä. Rethinking the evaluation of video summaries. In *CVPR*, 2019. 4, 6, 7
- [45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6
- [46] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Sumgraph: Video summarization via recursive graph modeling. In *ECCV*, 2020. 7
- [47] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021. 7, 12
- [48] Toby Perrett and Dima Damen. Ddlstm: dual-domain lstm for cross-dataset action recognition. In *CVPR*, 2019. 1, 2
- [49] 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 *ICML*, 2021. 3, 6, 13
- [50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020. 3
- [51] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 2013. 3
- [52] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *ECCV*, 2018. 2
- [53] Qian Ruan, Malte Ostendorff, and Georg Rehm. Histruct+: Improving extractive text summarization with hierarchical structure information. *arXiv:2203.09629*, 2022. 5
- [54] Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple pairwise ranking networks for personalized video summarization. In *ICCV*, 2021. 6
- [55] Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple pairwise ranking networks for personalized video summarization. In *ICCV*, 2021. 7
- [56] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv:2209.14792*, 2022. 6
- [57] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. 3
- [58] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 3
- [59] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015. 1, 2, 3, 4, 6
- [60] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv:1908.08530*, 2019. 3
- [61] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 3
- [62] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014. 3
- [63] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv:1908.07490*, 2019. 3
- [64] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 6
- [65] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv:2205.14100*, 2022. 3
- [66] Ke Wang, Hang Hua, and Xiaojun Wan. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *NeurIPS*, 2019. 8

- [67] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*, 2020. 3
- [68] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019. 3
- [69] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv:2104.14806*, 2021. 6
- [70] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv:2302.00402*, 2023. 6
- [71] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 3
- [72] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training. In *NeurIPS*, 2021. 3
- [73] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv:2205.01917*, 2022. 3
- [74] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021. 3
- [75] Yuan Yuan, Haopeng Li, and Qi Wang. Spatiotemporal modeling for video summarization using convolutional recurrent neural network. *IEEE Access*, 2019. 3
- [76] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *ECCV*, 2018. 12
- [77] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016. 2, 7
- [78] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, 2021. 3, 5
- [79] Yujia Zhang, Michael Kampffmeyer, Xiaoguang Zhao, and Min Tan. Dtr-gan: Dilated temporal relational adversarial network for video summarization. In *ACM TURC*, 2019. 3
- [80] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *ACM Multimedia*, 2017. 2
- [81] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *CVPR*, 2018. 2
- [82] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Tth-rnn: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Transactions on Industrial Electronics*, 2020. 2
- [83] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *AAAI*, 2018. 7
- [84] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 3
- [85] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 3
- [86] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 7
- [87] Wencheng Zhu, Yucheng Han, Jiwen Lu, and Jie Zhou. Relational reasoning over spatial-temporal graphs for video summarization. *IEEE Transactions on Image Processing*, 2022. 1
- [88] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 2020. 1, 2
- [89] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999. 6

A. Supplementary Overview

In the supplementary material, we first offer additional details on the proposed VideoXum dataset. Next, we elaborate on additional implementation details of the proposed framework. Furthermore, we provide an extended empirical analysis. Finally, we present more empirical results on the VideoXum dataset.

A.1. Dataset Details

A.1.1 Detailed Data Statistics

This section provides additional details on the dataset. Figure 5 shows the distributions of video lengths and normalized center timestamps of summarized video spans. As shown in Figure 5 (left), the video lengths vary greatly, ranging from 10 to 755 seconds, with 99.9% of them under 300 seconds. The average length is 124.2 seconds, and the median length is 121.6 seconds. Figure 5 (right) shows that the important clips are generally uniformly distributed throughout the video, with a mild peak at the beginning. Therefore, the VideoXum dataset does not suffer from temporal bias issues [31].

A.1.2 Reannotation Pipeline

This annotation pipeline aims to reannotate the video spans of ActivityNet Captions [29] and obtain an abridged version (ideally bounded to 15%) of video segments for better aligning with the sentence captions. In particular, crowd workers are given a video with several captions and the corresponding temporal annotations (i.e., pairs of start and end timestamps). For each caption, ten workers are assigned to reannotate the corresponding video segment and obtain ten shortened spans.

To ensure consistent annotations, we hired **40 crowd workers** to reannotate all **140,010 summarised video spans** over a period of **two months**. On average, each worker reannotated about **15 videos per hour**. To maintain high-quality annotations, we regularly reviewed the reannotated video spans and provided feedback to workers. Every 24 hours, we randomly evaluated 15% of an annotation batch for accuracy. If the acceptance rate of the sampled annotations reached 90%, we considered the entire annotation batch as passed; otherwise, we asked the workers to reannotate the batch.

A.2. Implementation Details

A.2.1 Additional Training Details

In this section, We provide additional training details about reproduction for reproducibility purposes.

Data Preprocessing. All video frames are first resized using bi-linear resampling to 224 pixels along the shorter

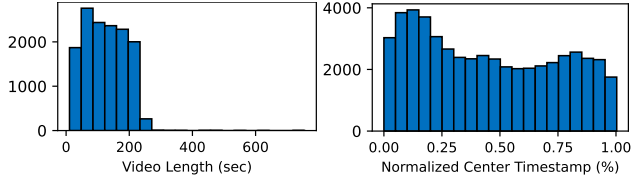


Figure 5: Distributions of video lengths (left) and normalized center timestamps of summarized video spans (right).

side. Next, a 224×224 center crop is applied to the resized frames. For each training batch, we add padding to all video sequences to make them the same length, enabling the videos to be processed in parallel and speed up the training process. In addition, the padding tokens are masked out during the self-attention calculation. Based on data statistics shown in Figure 5 (left), we set the maximum video length to 512, and frames exceeding the maximum length are truncated. For each text summary, we concatenate (dense) sentence captions in a video to construct a narrative paragraph [16, 47, 76]. According to data statistics shown in Figure 2 (right), we set the maximum generation length to 128 in the text summarization task.

Model Architecture. We employ ViT-B/16 [13] as the image encoder backbone with $N_{\text{vis}} = 12$ layers. The N_{tem} -layer Temporal Transformer (TT) follows the image encoder, where N_{tem} is 1. The temporal positional embeddings $\varepsilon_{\text{temp}}$ in Eq.(1) are also learnable. The video-sum decoder contains a Context Aggregation (CA) module capturing local context and a binary linear classifier. The CA module constructs a binary local attention map with window size $\epsilon = 5$. For the text-sum decoder, we adopt a variant of Transformer with $N_{\text{tex}} = 12$ layers, which replaces the bidirectional self-attention module with a causal self-attention module [13]. In addition, the prompt $\mathcal{T}_{\text{prompt}}$ of the text-sum decoder in Eq.(3) is set as “[DEC] a video of”.

Weight Initialization. To initialize the weights of our model, we employ a state-of-the-art pretrained vision-language model called BLIP [33]. The image encoder and the text-sum decoder of VTSUM-BLIP are initialized by pretrained BLIP_{CapFilt-L} [33]. Additionally, the Temporal Transformer and video-sum decoder are randomly initialized.

Finetuning. Due to limited computational resources, we finetuned all of the parameters in our proposed VTSUM-BLIP model, except for the image encoder. We adopt the AdamW [40] optimizer with an initial learning rate of 2×10^{-5} to optimize the model, and the weight decay is 5×10^{-2} . We train the VTSUM-BLIP model for 56 epochs with a batch size of 64 on 4 A100 GPUs. In addition, the weights of video-sum loss \mathcal{L}_v and text-sum loss \mathcal{L}_t are $\lambda_v = 15.0$ and $\lambda_t = 1.0$, respectively.

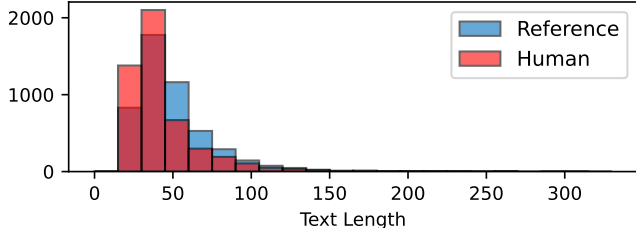


Figure 6: Distributions of text lengths of Reference and Human.

A.2.2 VT-CLIPScore

VT-CLIPScore aims to evaluate the text and video semantic consistency. We first finetune all of the parameters of the vanilla CLIP model [49] on our proposed VideoXum dataset, which enables the pretrained CLIP model to adapt to the VideoXum. In detail, we use the AdamW [40] optimizer with an initial learning rate of 2×10^{-6} and a weight decay of 5×10^{-2} . We finetune the CLIP model for 50 epochs with a batch size of 16 on 4 A100 GPUs. Based on the finetuned CLIP model, we calculate averaged CLIP similarity between video frames and text to derive the VT-CLIPScore.

A.3. Extended Empirical Analysis

A.3.1 Analysis for Human Performance on V2T-SUM

Human performance on our proposed three tasks can be regarded as an upper bound of each task. In Table 3, we can see that human performance outperforms our proposed model by a large margin on V2V-SUM and V2VT-SUM. However, on V2T-SUM, human performance does not exhibit a significant advantage over our model (especially on BLEU@4). To better understand this phenomenon, we examine several human-annotated examples and their corresponding references in Table 8, where “Human” indicates the human predictions on VideoXum *test* set and “Reference” denotes the corresponding ground truth. Both “Human” and “Reference” are human-annotated text summaries from ActivityNet Captions [29] validation set. As shown in Table 8, the examples demonstrate that summarizing a long video is inherently subjective, leading to varying text descriptions of the same content among different individuals. Figure 6 shows the statistical distribution of text summary length differences between the reference and human performance, highlighting the significant variability in annotators’ performance on the V2T-SUM task.

A.3.2 Impact of Local Window Size ε

For the V2V-SUM task, the local window size ε controls the context range of local self-attention. For $\varepsilon = 1$, the local self-attention module degrades to a multilayer perceptron (MLP). As ε increases to T (> 1), the local self-attention

Table 8: Selected V2T-SUM Examples.

Reference	Two young children are standing in line indoors in what looks like a living room. The little girl is standing closest to the hopscotch mat and she throws her toy onto the mat and then begins jumping until she meets the end of the mat then turns around and heads back to the point she started and her turn is over. The little boy goes next, and he throws the toy onto the mat and begins jumping to the end of the mat, then turns around and jumps back towards his starting point. The little girl steps in front of the boy and gets into motion to start another turn on the hopscotch mat.
Human	Two children stand in front of a mat. They throw something onto the mat. They take turns jumping across the mat. They pick up the item they threw on it.
Reference	People are in the river in a boat rushing downstream as the water splashes them. A mountain panorama is displayed onscreen. The 6 people row vigorously. We see them from the riverbed looking across, a farm building in the background. A woman talks to the camera. They take group selfies on dry land. People jump from a rock into the river. A man talks to the camera. A man points out the directions to other riders. Another woman talks to the camera. Food is being cooked for the party and plated to the attendees.
Human	A group of rafters are going through turbulent waters. A woman talks to the camera between shots of the rafters. They use their paddles as they go over the falls.

module upgrades to a global/regular self-attention module. To determine the impact of local window size ε , we perform an ablation study on the VSUM-BLIP model using the VideoXum *val* set. As shown in Figure 7, the optimal performance is obtained when ε is set to 5. The limited context would hamper the performance as the value of ε decreases below 5. On the other hand, when ε becomes larger than 5, the large context range would introduce irrelevant frames, resulting in suboptimal performance. Therefore, the performance of the V2V-SUM task is improved by carefully selecting appropriate local context information.

A.3.3 Impact of Temporal Transformer Layers

In this section, we conduct an ablation study on the number of Temporal Transformer layers N_{tem} using VideoXum *val* set. Table 9 indicates that altering N_{tem} does not significantly affect the performance for all three tasks. Therefore,

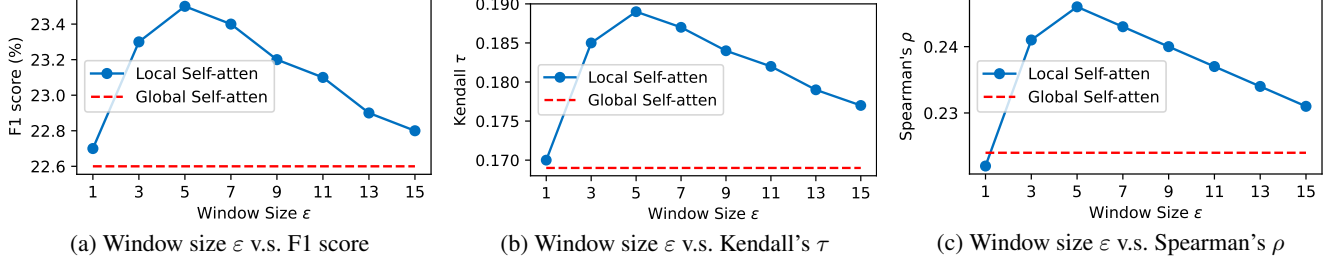


Figure 7: Impact of local window size ε .

Table 9: Impact of Temporal Transformer (TT) Layers N_{tem} on the VideoXum dataset *val* set. The F1 score, BLEU@4, METEOR, ROUGE-L, CIDEr, and VT-CLIPScore are shown in %.

Method	V2V-SUM			V2T-SUM				V2VT-SUM
	F1 score	Kendall	Spearman	BLEU@4	METEOR	ROUGE-L	CIDEr	VT-CLIPScore
VTSUM-BLIP (Base)	21.9	0.157	0.208	5.2	11.3	24.3	18.3	28.4
+ 1-layer TT	22.6	0.176	0.232	5.2	11.5	24.3	20.2	28.9
+ 2-layer TT	22.7	0.176	0.232	5.2	11.4	24.3	20.2	28.9
+ 3-layer TT	22.7	0.174	0.230	5.1	11.4	24.3	20.5	29.0
+ 4-layer TT	22.7	0.173	0.229	4.9	11.4	24.2	19.5	28.8

we set N_{tem} to 1.

A.4. Additional Qualitative Results

In this section, we present additional qualitative results in Figure 8 (on Page 15).

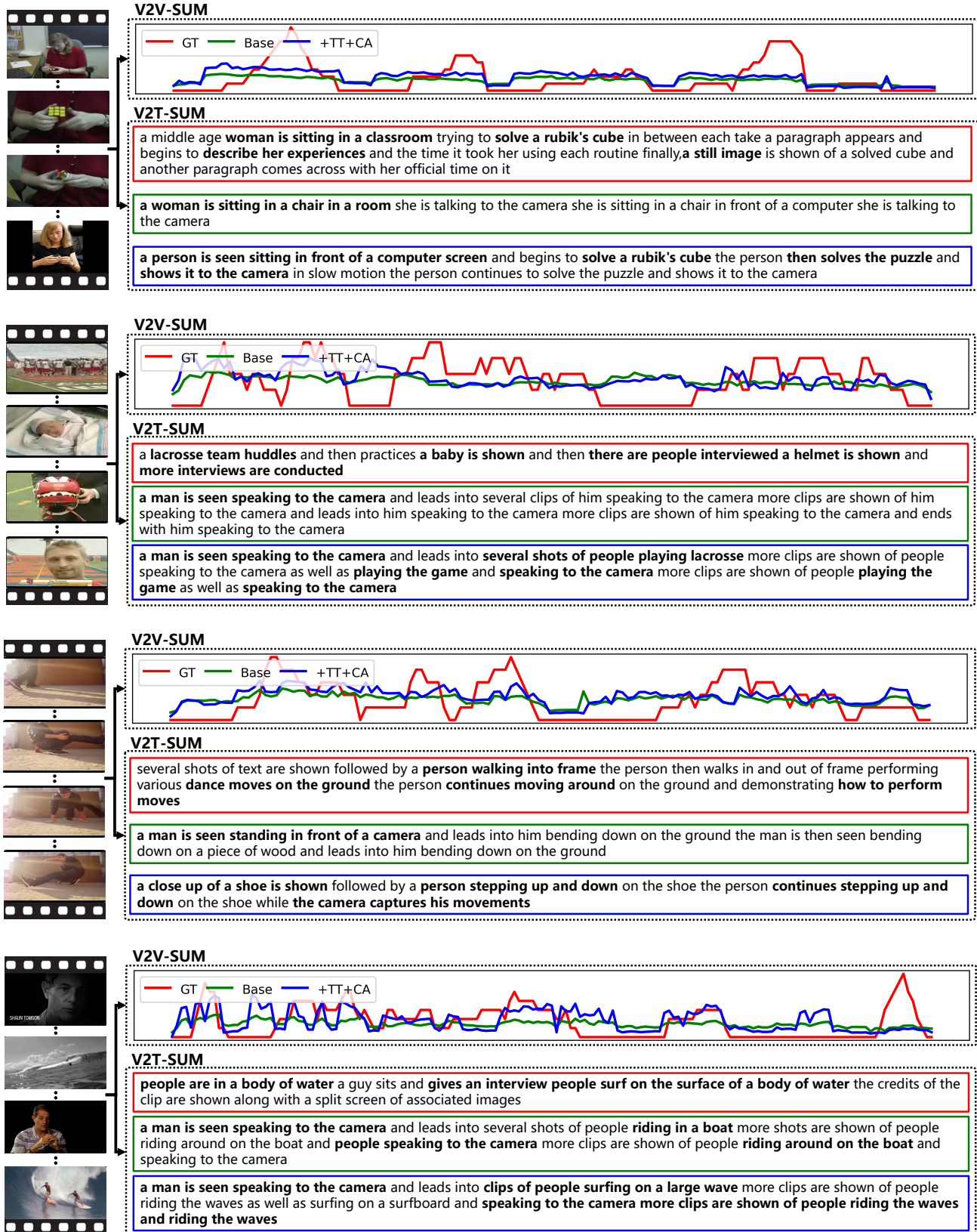


Figure 8: Additional example results of the generated video and text summaries across different baseline models. Red (both line and box) indicates the results of the ground truth. Green indicates the results of the VTSUM-BLIP (Base). Blue indicates the results of VTSUM-BLIP (+TT+CA).