Progressive Video Summarization via Multimodal Self-supervised Learning

Haopeng Li[1], Qiuhong Ke[3], Mingming Gong[2], Tom Drummond[1]
[1]School of Computing and Information Systems, The University of Melbourne
[2]School of Mathematics and Statistics, The University of Melbourne
[3]Department of Data Science & AI, Monash University
haopeng.li@student.unimelb.edu.au, {mingming.gong, tom.drummond}@unimelb.edu.au,
qiuhong.ke@monash.edu

Abstract

Modern video summarization methods are based on deep neural networks that require a large amount of annotated data for training. However, existing datasets for video summarization are small-scale, easily leading to over-fitting of the deep models. Considering that the annotation of large-scale datasets is time-consuming, we propose a multimodal self-supervised learning framework to obtain semantic representations of videos, which benefits the video summarization task. Specifically, the self-supervised learning is conducted by exploring the semantic consistency between the videos and text in both coarse-grained and fine-grained fashions, as well as recovering masked frames in the videos. The multimodal framework is trained on a newly-collected dataset that consists of video-text pairs. Additionally, we introduce a progressive video summarization method, where the important content in a video is pinpointed progressively to generate better summaries. Extensive experiments have proved the effectiveness and superiority of our method in rank correlation coefficients and F-score[1].

1. Introduction

Video summarization aims to generate a short version of a video by picking the most important frames or shots containing the main content of the original video, which greatly improves the efficiency of video browsing and retrieval. State-of-the-art video summarization methods are based on deep neural networks which model the dependencies between frames/shots and estimate their importance [42, 17, 16, 32, 41, 14]. However, existing datasets for video summarization are relatively small [12, 34], which easily leads to over-fitting of the deep models. Meanwhile, collecting a large-scale annotated dataset for video summarization is challenging and time-consuming, as multiple annotators need to provide frame/shot-level annotations to minimize subjectivity. When the annotated data are scarce, self-supervised learning has shown great power in boosting the performance of deep models in various scenarios, such as image retrieval [28], action recognition [1], and language understanding [8]. Encouraged by these successful stories, one may ask a natural question, "**Can self-supervised learning benefit video summarization?**"

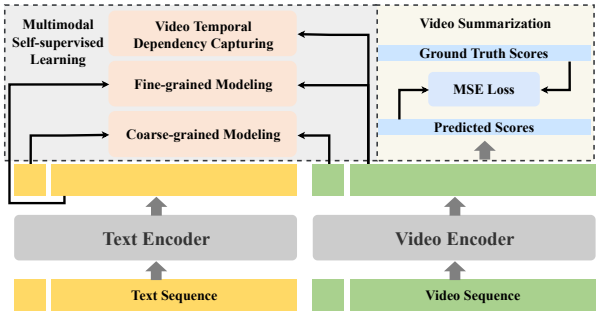


Figure 1. The proposed multimodal self-supervised framework for video summarization.

In this paper, we show that the answer to the above question is **YES**. Inspired by the semantic correlations between videos and their text information, we propose a new multimodal self-supervised framework for video summarization. In this framework, the multimodal correlations are captured from two aspects: 1) The coarse-grained modeling uses sequence-level representations of video and text to predict whether they have semantic correspondence; 2) The fine-grained modeling regards the video and the text as two sets and measures their distance using individual words and frames. Meanwhile, we also try to capture the temporal dependencies in the visual modality by modeling the relationship between the masked frame and the video. The proposed multimodal self-supervised framework is shown in Figure 1. To train our multimodal encoder, we collect a new dataset that consists of video-text pairs. Specifically, we

[1]The codes and dataset will be released soon.

first obtain several video *categories* and *search queries* from Google Trend[2]. We then collect the top searched videos whose duration is 3–20 minutes as well as their corresponding *titles* and *descriptions*. Finally, a dataset consisting of 3,081 **Y**ou**T**ube **V**ideo-**T**ext pairs (**YTVT**) is collected for the multimodal self-supervised learning.

After the self-supervised pretraining, a multimodal sequence encoder is obtained and further fine-tuned for the video summarization task. Existing video summarization methods [42, 17, 16, 14] are mainly based on a single-stage fashion where the videos are examined only once to generate the final summaries, which might be insufficient to pinpoint the important content. In this work, we propose a progressive video summarization method using the pretrained multimodal encoder, where the input sequence is refined by iteratively emphasizing the important content in a multi-stage fashion. Besides, we also describe how to incorporate the text information for video summarization.

Our contributions are summarized as follows:
1) We introduce multimodal self-supervised learning, where the multimodal correlations are modeled in both coarse-grained and fine-grained fashions. Meanwhile, the temporal dependencies in videos are captured by modeling the relationship between the masked frame and the whole video.
2) We collect a dataset of YouTube video-text pairs for multimodal self-supervised learning. The text of each video includes four types of information, depicting the video from the general category to the specific description.
3) Based on the pretrained video encoder, we propose a progressive video summarization method where the input video sequence is enhanced in a multi-stage fashion. We also incorporate text information for better video summarization.

2. Related Work

2.1. Video Summarization

Video summarization methods can be roughly classified into supervised methods and unsupervised ones. We review existing methods according to the category they belong to.

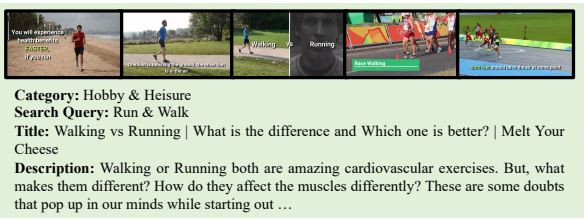
The unsupervised video summarization [24, 45, 14, 25, 24] relies on the criteria designed by human, such as representativeness [7, 27] and diversity [45]. Conventional machine learning algorithms such as clustering and dictionary learning were widely exploited in unsupervised methods. For instance, $L_{2,0}$-constrained sparse dictionary learning was used to address video summarization in [25]. Besides, unsupervised methods based on deep neural networks were presented in recent works. SGAN [24] used adversarial generative networks to generate summaries which were hard to discriminate from the original videos.

Most of the supervised methods are based on deep neural networks to model the temporal dependencies [40, 43, 44, 9, 42, 41, 32, 16], which requires human summaries for training. Numerous deep models were developed to capture the temporal dependencies in either local fashion or global fashion. For example, long short-term memory (LSTM) was exploited to model the video and predict the frame-level scores in vsLSTM/dppLSTM [40]. Furthermore, hierarchical adaptions of LSTM were proposed to address the issues of plain LSTM [43, 44]. Besides, attention models and graph models were exploited to capture the global dependencies. For instance, a sequence-graph structure was developed in RSGN [42], which models the frame-level dependencies and the shot-level ones successively. Considering the semantic correlations across videos, VJMHT [21] is developed based on a hierarchical Transformer. Nevertheless, most of the existing methods perform video summarization in a single-stage fashion where the videos are examined only once. In contrast, we propose progressive video summarization to iteratively refine the input and pinpoint the important content. Although SumGraph [30] also uses the recursive idea, our method is different to SumGraph in terms of motivation and methodology. Specifically, SumGraph recursively obtains a graph where the nodes are connected by stories instead of similarities, while our method iteratively emphasizes the important frames by inspecting the videos multiple times. Besides, SumGraph recursively refines the adjacent matrix in GCN, while our method reweights the input by the scores output from previous stage.

2.2. Multimodal Self-supervised Learning

Self-supervised learning has been widely used for the pretraining of deep models, which boosts their performance to a large extent in various fields [10, 37, 15, 39, 20, 23, 8]. Considering the semantic consistency in multimodal data, contrastive learning were exploited to model the correspondence among different modalities [19, 22, 3, 5, 4]. For instance, the consistency of videos and audios were leveraged to train the deep encoder in [19]. Besides, the semantic correlations between images and text were used to obtain semantic representations [22, 35, 46]. Such pretraining was proved effective in many tasks. Furthermore, the consistency among three modalities, video, audio, and text, were considered in [3, 2], by which a versatile network can be obtained. Self-supervised learning has been exploited for video summarization. Specifically, CLIP-It [26] uses the pretrained CLIP to extract frame features and the six-layer Transformer to obtain high performance. But it also brings large computation during testing. However, our method use traditional GoogLeNet features and a three-layer Transformer to achieves significant results (especially in rank-based evaluation) with reasonable computational cost. Besides, for self-supervised learning, we propose a framework that exploits the correspondence between modalities in both coarse-grained and fine-grained fashions, while considering

[2] https://trends.google.com/trends



Category: Hobby & Heisure
Search Query: Run & Walk
Title: Walking vs Running | What is the difference and Which one is better? | Melt Your Cheese
Description: Walking or Running both are amazing cardiovascular exercises. But, what makes them different? How do they affect the muscles differently? These are some doubts that pop up in our minds while starting out …

Figure 2. An example from the YTVT dataset. Five sampled frames and the text information of the video are presented.

Table 1. Statistics of YTVT. "Min/Max/Avg. Duration" represents the minimum/maximum/average duration of videos. "Avg. Title/Desc. Len." represents the average number of words in titles/descriptions.

Statistics	Result	Statistics	Result
#Videos	3,081	Min Duration	180s
#Categories	23	Max Duration	1,200s
#Queries	202	Avg. Duration	555.2s
Avg. Title Len.	9.7	Avg. Desc. Len.	98.6

the temporal dependencies in videos.

3. Multimodal Self-supervised Progressive Video Summarization

Deep learning has been popularly used for video summarization [17, 42, 44], yet most of the existing datasets [12, 34] are relatively small, resulting in over-fitting of the deep models. To resolve this issue, we explore self-supervised learning to improve video summarization. In this section, we propose multimodal **S**elf-**S**upervised **P**rogressive **V**ideo **S**ummarization (**SSPVS**). Specifically, we first introduce the collected dataset for the multimodal self-supervised learning. Then, we elaborate the framework for the multimodal self-supervised learning. We further present the progressive video summarization based on the pretrained encoders. Finally, we illustrate how to incorporate text information for better video summarization.

3.1. Dataset for Self-supervised Learning

Generally, there exist semantic correlations between videos and their associated text information such as titles and descriptions. Such correlations provide supervision that can be used to train a multimodal network in a self-supervised manner. This encourages the multimodal network to learn better representations of the video and text, which benefit the video summarization task. To learn the semantic correlation between the video and text information, video-text data are required. To this end, we first collect video data as well as their associated text information.

3.1.1 Data Collection

In this paper, we collect video data and their associated text information from YouTube. Specifically, we first obtain 23 video categories from Google Trend, such as *Autos & Vehicles* and *Beauty & Fitness*. For each category, we use its sub-categories as search queries and obtain search results on YouTube. For instance, the category of *Hobby & Leisure* has sub-categories such as *Cycling* and *Bowling*, and these sub-categories are used as search queries to collect more specific videos. Note that we manually eliminate trending queries such as *Gossip* and *Celebrity* because such videos contain few general scenarios. To make the self-supervised model robust to complex multimodal semantic correlations, we collect only the videos longer than 3 minutes to guarantee enough visual diversity of the data. Furthermore, considering the GPU memory limit, the videos longer than 20 minutes are also ruled out. The search results with required length are collected. Besides categories, we also collect the video-specific text information, including titles and descriptions. In summary, four types of text for each video are obtained: *category*, *search query*, *title*, and *description*. Figure 2 shows an example from YTVT[3].

3.1.2 Data Pre-processing

The collected text information cannot be directly used for self-supervised learning, because it contains a great deal of noise and irrelevant text which has no semantic meaning, especially in the *descriptions*. In this case, we first perform data cleaning by removing the noisy and meaningless text, including extra spaces, special symbols, non-Unicode characters, URLs, E-mails, etc. By this means, the remaining text is semantically related to the corresponding videos, which can be used for video-text joint modeling. Additionally, following the traditional pre-processing steps in NLP, we apply lemmatization to each word and lowercase all letters. Finally, a **YouTube**-based dataset of 3,081 **V**ideo-**T**ext pairs are collected for multimodal self-supervised pretraining, which is named as **YTVT**. Detailed statistics of YTVT (after pre-processing) are shown in Table 1.

3.2. Multimodal Self-supervised Pretraining

Given the video-text data, we investigate a multimodal network to exploit the correlation between videos and text information and model the temporal dependencies within videos. The framework is shown in Figure 3. Specifically, it consists of two unimodal encoders for the text information and the visual information, respectively. We explain the network structure and the learning objectives as follows.

[3]More examples can be found in the supplementary materials.

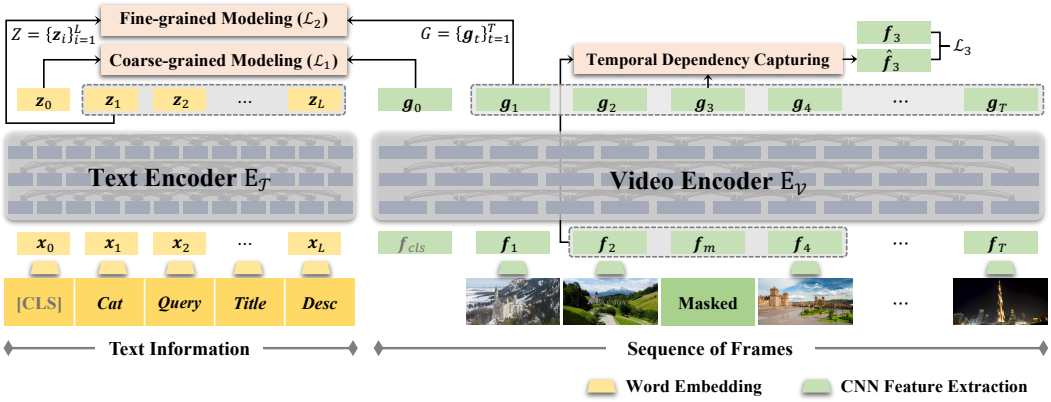


Figure 3. The overview of our multimodal self-supervised framework. It consists of a text encoder and a video encoder. The correspondence between the video and the text is modeled in both coarse-grained and fine-grained fashions. Additionally, the temporal dependencies in the video are captured by predicting the masked frame, considering the relationship between the masked frame and the whole video.

3.2.1 Network Structure

Text Encoder. For the input, four different types of text information of each video are considered, i.e., *category*, *search query*, *title*, and *description*, each consisting of a sequence of words. The four separate text sequences are combined to form a larger sequence, with a `[SEP]` token placing between every two types of text sequences to separate each other. The `[CLS]` token is prepended to the whole sequence to aggregate information from the text. Each word is converted into a vector by the pretrained word embedding module of BERT [8]. The dimension of word embeddings d_w is 768. The pretrained BERT model [8] is adopted to stabilize the text encoding. Formally, the encoded word sequence is denoted as $\{\boldsymbol{z}_i\}_{i=0}^{L}$, where $\boldsymbol{z}_0 \in \mathbb{R}^{d_w}$ is the representation of the whole text, $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_L \in \mathbb{R}^{d_w}$ are the encoded embeddings, and L is the number of words.

Video Encoder. Following [40, 24, 44], the output of the penultimate layer (pool 5) of the GoogLeNet [36] pretrained on ImageNet [33] is adopted as the frame feature, which is consistent with most of the video summarization methods for fair comparisons. The dimension of frame features d_f is 1,024. Similar to the `[CLS]` token for text encoding, a learnable feature ($\boldsymbol{f}_{cls} \in \mathbb{R}^{d_f}$) is prepended to the frame features to aggregate the temporal information in the video. The video encoder is a three-layer Transformer with random initialization. Formally, the encoded frame features are denoted as $\{\boldsymbol{g}_t\}_{t=0}^{T}$, where $\boldsymbol{g}_0 \in \mathbb{R}^{d_f}$ is the representation of the whole video sequence, $\boldsymbol{g}_1, \cdots, \boldsymbol{g}_T \in \mathbb{R}^{d_f}$ are the encoded frame features, and T is the number of frames.

3.2.2 Learning Objectives

In this work, we propose two types of learning objectives for self-supervised learning: 1) the semantic correlation between the video and the text, and 2) the temporal dependencies within videos, each of which is explained as follows.

Cross-modality Semantic Correspondence. Two sub-objectives are designed to model the correspondence between the video and the text: the coarse-grained modeling and the fine-grained modeling. The coarse-grained modeling exploits sequence-level representations to capture the semantic correlation. For each video-text pair during training, the text information is replaced with that of another video with a probability 50% to generate the negative pair. Formally, given the representations of the video and the text, $\boldsymbol{g}_0, \boldsymbol{z}_0$, we use a three-layer perceptron $\text{MLP}_{cls}(\cdot)$ to predict whether the video and the text are corresponding, i.e., $p_c = \text{MLP}_{cls}([\boldsymbol{g}_0, \boldsymbol{z}_0])$, where $[\cdot,\cdot]$ represents concatenation. Following previous self-supervised learning [22, 35], binary cross entropy is used as the sub-objective,

$$\mathcal{L}_1 = -(y \log(p_c) + (1-y)\log(1-p_c)), \tag{1}$$

where y is the binary label indicating whether the video-text pair is corresponding or not. The coarse-grained modeling focuses on sequence-level representations which contains only the global information. However, the local information in the video and the text are also of great importance to video understanding. Motivated by this, we propose fine-grained modeling for the correspondence between the video and the text. Formally, given the set of the encoded frames $\mathcal{G} = \{\boldsymbol{g}_t\}_{t=1}^{T}$ and the set of the encoded word embeddings $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^{L}$, we first measure the distance between the two sets using Hausdorff distance d_H as follows,

$$d_H(\mathcal{G}, \mathcal{Z}) = \max\{d(\mathcal{G}, \mathcal{Z}), d(\mathcal{Z}, \mathcal{G})\}, \tag{2}$$

$$d(\mathcal{G}, \mathcal{Z}) = \max_t \min_i \left\| \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|_2} - \frac{\text{MLP}_\mathcal{T}(\boldsymbol{z}_i)}{\|\text{MLP}_\mathcal{T}(\boldsymbol{z}_i)\|_2} \right\|_2, \tag{3}$$

$$d(\mathcal{Z}, \mathcal{G}) = \max_i \min_t \left\| \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|_2} - \frac{\text{MLP}_\mathcal{T}(\boldsymbol{z}_i)}{\|\text{MLP}_\mathcal{T}(\boldsymbol{z}_i)\|_2} \right\|_2, \tag{4}$$

where we use a two-layer perceptron $\mathrm{MLP}_{\mathcal{T}}(\cdot)$ to map the encoded word embeddings into the visual space. Based on the distance of the two sets, we exploit the contrastive loss to pull the corresponding video-text sets together and push the unmatched sets away, i.e.,

$$\mathcal{L}_2 = y d_H^2 + (1-y)\max\{0, m - d_H^2\}, \tag{5}$$

where m is a pre-defined margin. In this sub-objective, instead of using a holistic representation for the video or the text, we model the semantic correlation between the video and the text by inspecting individual frames and words. By this means, the framework can discover more fine-grained cross-modal information for video understanding.

Temporal Dependencies in Videos. We also capture the temporal dependencies in videos. Similar to the training of BERT, we randomly mask a frame and require the model to recover it. Specifically, we replace a randomly selected frame with a learnable feature ($\boldsymbol{m} \in \mathbb{R}^{d_f}$). Instead of predicting the masked frame using the encoded masked feature as most methods, we recover the frame by considering the temporal dependencies between the masked frame and whole video, i.e., whether the masked frame is a smooth transition or an abrupt one. To this end, a two-layer perceptron $\mathrm{MLP}_s(\cdot)$ is applied to the encoded masked feature to predict the probability of being a smooth transition. Assuming the t-th frame is masked, the probability is computed as $p_s = \mathrm{MLP}_s(\boldsymbol{g}_t)$, where $\boldsymbol{g}_t$ is the encoded feature of $\boldsymbol{m}$. Then, the masked frame is recovered from two aspects: 1) If it is a smooth transition, we recover it by using only its neighbors (local information) with an one-layer Transformer ($\mathrm{T}_{\mathcal{V}}$) and a linear projection as follows,

$$\boldsymbol{R} = \mathrm{T}_{\mathcal{V}}([\boldsymbol{f}_{t-k}, \cdots, \boldsymbol{m}, \cdots, \boldsymbol{f}_{t+k}]^{\mathrm{T}} + \boldsymbol{E}^{\mathcal{V}}_{pos}), \tag{6}$$

$$\hat{\boldsymbol{f}}_t^1 = \boldsymbol{W}_1 \boldsymbol{R}_c \tag{7}$$

where k is a pre-defined window radius, $\boldsymbol{E}^{\mathcal{V}}_{pos} \in \mathbb{R}^{(2k+1)\times d_f}$ is the positional encoding, $\boldsymbol{R}_c \in \mathbb{R}^{d_f}$ is the feature in $\boldsymbol{R}$ corresponding to $\boldsymbol{m}$, and $\boldsymbol{W}_1 \in \mathbb{R}^{d_f \times d_f}$ is a learnable parameter. 2) If the masked frame is an abrupt transition (which means only the local information is insufficient for inferring the masked frame), we use $\boldsymbol{g}_t$ to recover it, as $\boldsymbol{g}_t$ contains global information of the video. Specifically, a simple linear projection is applied to $\boldsymbol{g}_t$ to predict the masked frame, i.e., $\hat{\boldsymbol{f}}_t^2 = \boldsymbol{W}_2 \boldsymbol{g}_t$, where $\boldsymbol{W}_2 \in \mathbb{R}^{d_f \times d_f}$ is a learnable parameter. Considering the two situations, the masked frame is recovered by combining $\hat{\boldsymbol{f}}_t^1$ and $\hat{\boldsymbol{f}}_t^2$, i.e.,

$$\hat{\boldsymbol{f}}_t = p_s \hat{\boldsymbol{f}}_t^1 + (1-p_s)\hat{\boldsymbol{f}}_t^2. \tag{8}$$

The loss is defined as the mean squared error between the masked frame and the recovered one, i.e.,

$$\mathcal{L}_3 = \frac{1}{d_f} \left\| \hat{\boldsymbol{f}}_t - \boldsymbol{f}_t \right\|_2^2. \tag{9}$$

Note that there is no label for p_s, and we only apply supervision for the recovered frames, which forces the model to adaptively re-weight the predictions from two scenarios.

In summary, the training of the multimodal framework involves three loss functions, and we use their combination for self-supervised learning, i.e.,

$$\mathcal{L}_{SSL} = \mathcal{L}_1 + \alpha \mathcal{L}_2 + \beta \mathcal{L}_3, \tag{10}$$

where α, β are hyper-parameters to balance the three terms.

3.3. Progressive Video Summarization

In this section, we describe how to perform summarization using the pretrained encoders. Existing methods perform video summarization in the single-stage fashion where the videos are examined only once, which might be insufficient to pinpoint the important content. To address this limitation, we propose progressive video summarization. As shown in Figure 4(a), the framework is a stack of multiple models with the identical architecture. Each model is referred to as a *stage* whose structure is shown in Figure 4(b).

More specifically, the input of the n-th stage is computed as the weighted enhancement of the input of the previous stage based on the output of the previous stage, i.e.,

$$\boldsymbol{F}^n = \boldsymbol{F}^{n-1} * \boldsymbol{s}^{n-1} + \boldsymbol{F}^{n-1}, \tag{11}$$

where $\boldsymbol{F}^n = [\boldsymbol{f}_1^n, \cdots, \boldsymbol{f}_T^n]^{\mathrm{T}} \in \mathbb{R}^{T \times d_f}$ represents the sequence of input features at the n-th stage. $\boldsymbol{f}_t^n \in \mathbb{R}^{d_f}$ denotes the feature at the t-th time step of the sequences $\boldsymbol{F}^n$. $\boldsymbol{s}^{n-1} = \left[s_1^{n-1}, \cdots, s_T^{n-1}\right]^{\mathrm{T}} \in \mathbb{R}^T$ is a sequence of the frame scores output by the $(n-1)$-th stage. $s_t^{n-1} \in [0,1]$ is a scalar denoting the score at the t-th time step of $\boldsymbol{s}^{n-1}$. T represents the number of frames in the sequence. $*$ represents row-wise multiplication. For the first stage, $\boldsymbol{F}^0$ is initialized as the original frame features extracted by the pretrained CNN, and $\boldsymbol{s}^0 := \boldsymbol{0}$.

The underlying motivation of formulating $\boldsymbol{F}^n$ as in Eq. (11) is to iteratively refine the video sequence by emphasizing the important content. We find that, even though the video encoder receives scaled versions of the input sequence in the n-th stage ($n > 1$) (due to the residual connection), it can still model the sequence properly and performs much better than the formulation without the residual connection (**see supplementary materials for details**).

At the n-th stage, the input feature $\boldsymbol{F}^n$ is encoded by the pretrained video encoder ($\mathrm{E}_{\mathcal{V}}$), i.e.,

$$\boldsymbol{G}^n = \mathrm{E}_{\mathcal{V}}(\boldsymbol{F}^n + \boldsymbol{E}^{\mathcal{V}}_{pos}), \tag{12}$$

where $\boldsymbol{G}^n = [\boldsymbol{g}_1^n, \cdots, \boldsymbol{g}_T^n]^{\mathrm{T}} \in \mathbb{R}^{T \times d_f}$ is the sequence of encoded features. $\boldsymbol{g}_t^n \in \mathbb{R}^{d_f}$ denotes the feature at t-th time step of $\boldsymbol{G}^n$. $\boldsymbol{E}^{\mathcal{V}}_{pos} \in \mathbb{R}^{T \times d_f}$ is the positional encoding for

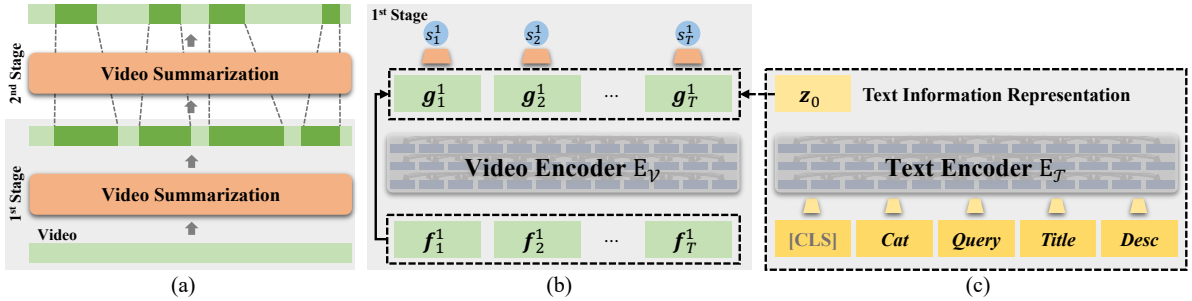


Figure 4. (a) The overview of the proposed progressive video summarization of two stages, where the structure and process of each stage is identical. (b) The details of video summarization in the first stage. (c) The details of the text information encoding.

the video sequence. Note that the video encoders in different stages are identical and share the parameters. Containing the global temporal information of the sequence, $\boldsymbol{G}^n$ is then leveraged to predict the frame-level importance scores. Additionally, a residual connection is applied before the linear projection to improve training stability [13], i.e.,

$$\boldsymbol{s}^n = \sigma((\boldsymbol{G}^n + \boldsymbol{F}^n)\boldsymbol{W}^n + b^n), \tag{13}$$

where $\boldsymbol{s}^n \in \mathbb{R}^T$ is the output sequence of the frame scores at the n-th stage, $\boldsymbol{W}^n \in \mathbb{R}^{d_f}, b^n \in \mathbb{R}$ are the learnable parameters of the n-th stage, and $\sigma(\cdot)$ is the Sigmoid function.

The final frame scores are computed by taking into consideration the score sequences output by all stages, i.e.,

$$\boldsymbol{s}^* = \boldsymbol{s}^1 \odot \boldsymbol{s}^2 \odot \cdots \odot \boldsymbol{s}^N \in \mathbb{R}^T, \tag{14}$$

where $\odot$ is element-wise multiplication, and N is the number of stages. **More manners to compute the final scores can be found in the supplementary materials**.

Optimization and Summary Generation. We use the the ground truth frame-level importance scores provided in the datasets to train the proposed video summarization framework. Formally, given the ground truth frame scores $\boldsymbol{s}_{gt} \in \mathbb{R}^T$ and the predicted frame scores $\boldsymbol{s}^*$ of a video, the mean square error is exploited as the loss function, i.e.,

$$\mathcal{L}_{VS} = \frac{1}{T}\left\|\boldsymbol{s}^* - \boldsymbol{s}_{gt}\right\|_2^2. \tag{15}$$

To generate summaries, we follow [40, 24, 45] and select shots to maximize the total score, with the constraint that the length of the summary is less than 15% of that of the original video. Kernel-based Temporal Segmentation (KTS) [31] is used to segment the videos into shots. The score of a shot is the average score of the frames within it.

3.4. Video Summarization with Text Information

With the pretrained text encoder, we can optionally incorporate text information into video summarization. As shown in Figure 4(c), given the text information, the `[CLS]` token is prepended to the whole sequence, and the pretrained word embedding model is applied to convert the text to a sequence of word embeddings $\boldsymbol{X} = [\boldsymbol{x}_0, \cdots, \boldsymbol{x}_L]^{\mathrm{T}} \in \mathbb{R}^{(L+1)\times d_w}$, where L and d_w are the length of the word sequence and the dimension of the word embedding. The sequence is then encoded by the pretrained text encoder ($\mathrm{E}_{\mathcal{T}}$),

$$\boldsymbol{Z} = \mathrm{E}_{\mathcal{T}}(\boldsymbol{X} + \boldsymbol{E}^{\mathcal{T}}_{pos}), \tag{16}$$

where $\boldsymbol{Z} = [\boldsymbol{z}_0, \cdots, \boldsymbol{z}_L]^{\mathrm{T}} \in \mathbb{R}^{(L+1)\times d_w}$ is the sequence of the encoded word embeddings. $\boldsymbol{E}^{\mathcal{T}}_{pos} \in \mathbb{R}^{(L+1)\times d_w}$ is the positional encoding for the word sequence. Finally, $\boldsymbol{z}_0 \in \mathbb{R}^{d_w}$, the encoded feature of the `[CLS]` token, is regarded as the feature of the text information, which is fused into the visual modality for video summarization as follows,

$$\boldsymbol{s}^1 = \sigma((\boldsymbol{G}^1 + \boldsymbol{F}^1 + \mathrm{MLP}_{\mathcal{T}}(\boldsymbol{z}_0))\boldsymbol{W}^1 + b^1). \tag{17}$$

Note that the text representation is used in only the first stage. By this means, the frame-level scores are predicted by taking into consideration not only the visual information but also the associated text information.

4. Experiments

We conduct experiments to verify the effectiveness of our method. The experimental settings are first explained. We then compare our method with other video summarization methods and video-text pretraining methods. Finally, we perform ablation studies to demonstrate the impact of our contributions. Additionally, **the predicted scores are visualized in the supplementary material**.

4.1. Experimental Settings

4.1.1 Datasets for Video Summarization

We use SumMe [12] and TVSum [34] datasets for video summarization. For SumMe, we regard their video names in the dataset as *search query* of the text information and the other three types of text information are left empty. For TVSum, the video titles are provided in the dataset, and the 10 categories in TVSum are regarded as *search query* of the

text information. Besides, we re-collect the *category* information from YouTube (if not available then left emtpy), and *description* is left empty. Apart from SumMe and TVSum, YouTube [7] and OVP[4] are used for the augmented setting and the transfer setting [40, 45].

4.1.2 Implementation Details

Multimodal Self-supervised Pretraining. The videos in YTVT are sub-sampled to 2 FPS to reduce temporal redundancy, which is consistent with SumMe and TVSum. For the video encoder ($\mathrm{E}_{\mathcal{V}}$) and the Transformer ($\mathrm{T}_{\mathcal{V}}$) in Eq. (6), the number of heads in the multi-head attention is set to 8. The dimension of the feed-forward network is set to 4,096. To increase the generalization ability, we randomly crop 256 frames from each video for training. The uncased base version of the BERT model is used as the text encoder ($\mathrm{E}_{\mathcal{T}}$). To deal with the inconsistency of text lengths in training, we set the maximum lengths of *category*, *search query*, *title*, and *description* to 3, 3, 10, and 50. For those text sequences shorter than the requirements, they are padded with the `[PAD]` tokens in the end. The margin m in Eq. (5) is set to $\sqrt{2}$. The windows radius k in Eq. (6) is set to 4. We set $\alpha = 1, \beta = 5$ in Eq. (10). The framework is optimized by Adam [18] with batch size 8 and learning rate 10^{-6}.
Progressive Video Summarization. The framework is trained by Adam for 40 epochs with batch size 4 and learning rate 10^{-5}. The maximum length of the videos is set to 512 frames by random temporal cropping. The videos shorter than the requirement are padded with zeros in the end. When training the model with text information, considering the average length of each type of text, we set the maximum lengths of *category*, *search query*, *title*, and *description* to 1, 3, 10, and 15. Following [14, 16, 45], five-fold cross-validation is performed.

4.1.3 Evaluation Metrics

F-score. F-score measures the overlap between the generated video summary and the human summary. Specifically, given the generated video summary $\mathcal{V}_s$ and the human summary $\mathcal{V}_{gt}$, the precision P and recall R are computed as $P = \frac{|\mathcal{V}_s \cap \mathcal{V}_{gt}|}{|\mathcal{V}_s|}, R = \frac{|\mathcal{V}_s \cap \mathcal{V}_{gt}|}{|\mathcal{V}_{gt}|}$. The F-score F is computed as the harmonic average of P and R, i.e., $F = \frac{2PR}{P+R}$.
Rank-based Evaluation. Rank-based evaluation is proposed [29] to address the limitations of F-score. Specifically, given the predicted frame-level scores and the scores annotated by human, two rank correlation coefficients, Kendall's τ and Spearman's ρ, are used as the primary comparison metrics in the experiments. For the videos with multiple sets of annotations, the average coefficients are taken as the final results, and the same goes for F-score.

[4]Open Video Project: https://open-video.org

Table 2. The results (Kendall's τ and Spearman's ρ) on SumMe and TVSum. The methods in the first row are unsupervised methods, while those in the second row are supervised methods.

Methods	SumMe		TVSum	
	τ	ρ	τ	ρ
SGAN [24]	—	—	0.024	0.032
WS-HRL [6]	—	—	0.078	0.116
DRDSN [45]	0.047	0.048	0.020	0.026
RSGN_u [42]	0.071	0.073	0.048	0.052
dppLSTM [40]	—	—	0.042	0.055
CSNet_s [16]	—	—	0.025	0.034
GLRPE [17]	—	—	0.070	0.091
SumGraph [30]	—	—	0.094	0.138
HSA [44]	0.064	0.066	0.082	0.088
RSGN [42]	0.083	0.085	0.083	0.090
SSPVS	0.178	0.240	0.177	0.233
SSPVS+Text	**0.192**	**0.257**	**0.181**	**0.238**

4.2. Comparisons with the State of the Art

4.2.1 Comparisons of Rank Correlation Coefficients

We compare our methods with the state of the art using the rank correlation coefficients, Spearman's ρ and Kendall's τ. The results are shown in Table 2.

As shown in Table 2, SSPVS outperforms existing methods to a large extent on both datasets, which means the proposed method can model the relative importance among frames more accurately than previous works. Additionally, by including the text information in the summarization process, the performance is further improved on both datasets.

4.2.2 Comparisons of F-Score

We also compare our method with previous works in the widely-used F-score. The results are shown in Table 3. Since the text information of the videos in OVP and YouTube is not collected, the results of SSPVS+Text in the augmented setting and the transfer setting are not reported.

As shown in Table 3, SSPVS outperforms most of the compared methods, which proves its effectiveness in pinpointing the important shots. We also find that our method is inferior to DSNet [47] and MSVA [11]. However, DSNet formulates video summarization as temporal detection and requires complex training and testing strategy, which is less applicable. As for MSVA, it uses extra C3D-based features, which is effective but also brings large computation.

4.2.3 Comparisons of Self-supervised Models

We also compare the proposed self-supervised learning method with other frameworks, including VideoBERT [35], VideoClip [38], and VATT [2]. For fair comparisons, we

Table 3. F-score in different settings on SumMe and TVSum. The methods in the first row are unsupervised methods, while those in the second row are supervised methods. *Can*/*Aug*/*Tran* represents the canonical/augmented/transfer setting.

Methods	SumMe			TVSum		
	Can	*Aug*	*Tran*	*Can*	*Aug*	*Tran*
SGAN [24]	0.387	0.417	—	0.508	0.589	—
DRDSN [45]	0.414	0.428	0.424	0.576	0.584	0.578
ACGAN [14]	0.460	0.470	0.445	0.585	0.589	0.578
WS-HRL [6]	0.436	0.445	—	0.584	0.585	—
vsLSTM [40]	0.376	0.416	0.407	0.542	0.579	0.569
$SGAN_s$ [24]	0.417	0.436	—	0.563	0.612	—
H-RNN [43]	0.421	0.438	—	0.579	0.619	—
HSA [44]	0.423	0.421	—	0.587	0.598	—
re-S2S [41]	0.425	0.449	—	0.603	**0.639**	—
S-FCN [32]	0.475	**0.511**	0.441	0.568	0.592	**0.582**
$CSNet_s$ [16]	0.486	0.487	0.441	0.585	0.571	0.574
DSNet [47]	0.502	0.507	**0.465**	**0.621**	**0.639**	**0.594**
MSVA [11]	**0.534**	—	—	0.615	—	—
SSPVS	0.487	0.504	0.458	0.603	0.618	0.578
SSPVS+Text	0.507	—	—	0.604	—	—

Table 4. The results (Kendall's τ and Spearman's ρ) of different self-supervised methods.

Methods	SumMe		TVSum	
	τ	ρ	τ	ρ
Baseline	0.137	0.187	0.141	0.185
VATT [2]	0.137	0.185	0.143	0.188
VideoBERT [35]	0.142	0.191	0.145	0.190
VideoClip [38]	0.139	0.188	0.148	0.195
SSPVS	**0.154**	**0.207**	**0.151**	**0.199**

pretrain the compared models on YTVT with the same form of inputs as our method. Specifically, instead of using images (or their quantizations) as video input, we use the extracted VGG features of frames. Besides, the acoustic modality in VATT is discarded. Then the pretrained video encoders are fine-tuned for video summarization, where the progressive mechanism and text information are not applied. The baseline is our model without pretraining. We report the results of rank-based evaluation in Table 4. As the results show, VATT pretraining reveals little impact on video summarization, while VideoBERT improves the performance marginally on both datasets. Besides, VideoClip improves the results on TVSum significantly. As for our framework, it improves the results significantly, which indicates the superiority of our method over other self-supervised methods in video summarization. Additionally, **the impact of each proposed self-supervised loss ($\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$) is shown in the supplementary material**.

4.3. Ablation Study

We conduct ablation studies to demonstrate the effectiveness of the multimodal self-supervised pretraining, the progressive summarization, and the summarization with text information. Considering the computational complexity and the GPU memory limit, we set the number of stages to 1–4. For each number of stages, three models are evaluated: the base model (without pretraining and text information), the model with pretraining, and the model with pretraining and text information. The results are shown in Table 5.

Table 5. The results of ablation studies in rank-based evaluation.

#Stages	Pretrain	Text	SumMe		TVSum	
			τ	ρ	τ	ρ
1			0.137	0.187	0.141	0.185
	✓		0.154	0.207	0.151	0.199
	✓	✓	0.159	0.212	0.157	0.206
2			0.140	0.189	0.159	0.209
	✓		0.162	0.217	0.161	0.212
	✓	✓	0.174	0.235	0.163	0.214
3			0.166	0.224	0.162	0.212
	✓		0.178	0.240	0.172	0.226
	✓	✓	**0.192**	**0.257**	0.173	0.228
4			0.145	0.198	0.163	0.214
	✓		0.172	0.231	0.177	0.233
	✓	✓	0.175	0.237	**0.181**	**0.238**

As the results show, self-supervised pretraining improves the performance significantly for each number of stages. We find that the impact on SumMe is greater than that on TVSum. We believe the reason is that SumMe consists of less training videos than TVSum, and so the pretraining can be more useful on SumMe. Additionally, exploiting text information also benefits video summarization, but the impact becomes less obvious when the number of stages increases. As for the progressive mechanism, with the increase of the number of stages, the performance is improved on both datasets. The best performance is reached at three stages on SumMe and at four stages on TVSum.

5. Conclusion

We have successfully incorporated video summarization into the self-supervised learning framework which leverages the coarse-grained and fine-grained semantic consistency between the video and the text as well as the temporal dependencies in videos. Based on the pretrained encoders, we have developed progressive video summarization, where the input sequences are refined in a multi-stage fashion and the text information can also be leveraged. Extensive experiments have verified the effectiveness of our contributions. Compared with previous works, our method have achieved state-of-the-art performance in rank-based evaluation.

Acknowledgement

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200. MG was supported by ARC DE210101624.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019.

[2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelovic, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *NeurIPS*, 2(6):7, 2020.

[4] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.

[5] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018.

[6] Yiyan Chen, Li Tao, Xueting Wang, and Toshihiko Yamasaki. Weakly supervised video summarization by hierarchical reinforcement learning. In *Proceedings of the ACM Multimedia Asia*, pages 1–6. 2019.

[7] 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*, 32(1):56–68, 2011.

[8] 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.

[9] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Asian Conference on Computer Vision*, pages 39–54. Springer, 2018.

[10] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

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

[12] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Xufeng He, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Unsupervised video summarization with attentive conditional generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2296–2304, 2019.

[15] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[16] 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*, volume 33, pages 8537–8544, 2019.

[17] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *European Conference on Computer Vision, ECCV 2020*. Springer, 2020.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

[20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[21] Haopeng Li, Qiuhong Ke, Mingming Gong, and Rui Zhang. Video joint modelling based on hierarchical transformer for co-summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2022.

[22] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[23] 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.

[24] 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.

[25] Shaohui Mei, Genliang Guan, Zhiyong Wang, Mingyi He, Xian-Sheng Hua, and David Dagan Feng. L 2, 0 constrained sparse dictionary selection for video summarization. In *2014 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2014.

[26] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *Advances in Neural Information Processing Systems*, 34:13988–14000, 2021.

[27] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Automatic video summarization by graph modeling. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 104–109. IEEE, 2003.

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[29] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7596–7604, 2019.

[30] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Sumgraph: Video summarization via recursive graph modeling. In *European Conference on Computer Vision*, pages 647–663. Springer, 2020.

[31] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[32] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 347–363, 2018.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[34] 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.

[35] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[37] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1329–1338, 2017.

[38] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[39] 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.

[40] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[41] 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.

[42] Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. Reconstructive sequence-graph network for video summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[43] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 863–871, 2017.

[44] 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.

[45] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[46] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020.

[47] 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.