Deep Attentive Video Summarization With Distribution Consistency Learning

Zhong Ji, *Member, IEEE*, Yuxiao Zhao, Yanwei Pang, *Senior Member, IEEE*, Xi Li, and Jungong Han

***Abstract*— This article studies supervised video summarization by formulating it into a sequence-to-sequence learning framework, in which the input and output are sequences of original video frames and their predicted importance scores, respectively. Two critical issues are addressed in this article: short-term contextual attention insufficiency and distribution inconsistency. The former lies in the insufficiency of capturing the short-term contextual attention information within the video sequence itself since the existing approaches focus a lot on the long-term encoder–decoder attention. The latter refers to the distributions of predicted importance score sequence and the ground-truth sequence is inconsistent, which may lead to a suboptimal solution. To better mitigate the first issue, we incorporate a self-attention mechanism in the encoder to highlight the important keyframes in a short-term context. The proposed approach alongside the encoder–decoder attention constitutes our deep attentive models for video summarization. For the second one, we propose a distribution consistency learning method by employing a simple yet effective regularization loss term, which seeks a consistent distribution for the two sequences. Our final approach is dubbed as Attentive and Distribution consistent video Summarization (ADSum). Extensive experiments on benchmark data sets demonstrate the superiority of the proposed ADSum approach against state-of-the-art approaches.**

***Index Terms*— Distribution consistency, self-attention, sequence-to-sequence (Seq2Seq) learning, video summarization.**

I. Introduction

BY CONDENSING a video into a concise yet comprehensive summary, video summarization provides an efficient and effective video browsing and thus increases understanding of video contents. It can be widely used in applications of online video management, interactive browsing and searching, and intelligent video surveillance [1]–[5]. Due to its great significance, video summarization has been a crucially urgent task, especially in the era of big video data.

Manuscript received May 11, 2019; revised October 26, 2019 and February 4, 2020; accepted April 25, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61771329, Grant 61472273, and Grant 61632018. *(Corresponding author: Zhong Ji.)*

Zhong Ji, Yuxiao Zhao, and Yanwei Pang are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: jizhong@tju.edu.cn; 2117234097@tju.edu.cn; pyw@tju.edu.cn).

Xi Li is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, and also with the Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China (e-mail: xilizju@zju.edu.cn).

Jungong Han is with the Data Science Group, University of Warwick, Coventry CV4 7AL, U.K. (e-mail: jungonghan77@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2991083

In recent years, some important signs of progress have been made in the research of supervised learning-based video summarization. To explicitly learn the summarization capability from human, it seeks supervised learning methods by employing videos and their corresponding human-created summary ground truths as training data. Many supervised learning methods, such as supervised subset selection [6], SVM [7], multiple objective optimization [1], and sequence-to-sequence (Seq2Seq) learning [2], [3], [8], have been developed.

Among them, Seq2Seq learning-based approach pioneers a promising direction [2], [3], [8]–[10]. It formulates the video summarization as a structure prediction problem on sequential data, where the input is the original video frame sequence and the output is the predicted importance score sequence of video frames. Based on these predicted importance scores, the keyframe/keyshot can be determined. This line of approach advocates the use of long-short term memory (LSTM) [11] as the encoder to map the visual sequence to a fixed dimensional vector. LSTM stems from recurrent neural network (RNN) architecture and is particularly good at modeling the variable-range temporal dependences among video frames. It is a crucial property for generating a meaningful summary since one of the challenges for video summarization is to effectively model the long-term temporal contextual information. Meanwhile, multilayer perceptron (MLP) and LSTM are usually selected as decoders to decode the vector into an importance score sequence.

One of the pioneering studies can be traced back to vsLSTM approach [2], where a bidirectional LSTM (BiLSTM) is exploited to encode the variable range dependence in a video, and an MLP is employed to combine the hidden states of LSTM layers and the visual features to indicate the likelihoods of whether the frames should be chosen in the summary. AVS [3], $\text{SASUM}_{\text{sup}}$ [8] and $\text{SUM-GAN}_{\text{sup}}$ [9] follow this structure by replacing MLP with LSTM in the encoder. Specifically, as our baseline method, AVS [3] further exploits an encoder–decoder attention mechanism in this framework to assign different importance weights to different input frames for strengthening their differences. By doing so, the long-term contextual attention is strengthened.

While the results are encouraging, the short-term contextual attention on the input frames is neglected. Similar to the long-term contextual attention, we emphasize that the short-term contextual attention should also be highlighted. This is because the frames in a video clip contribute unevenly to the encoded vector, especially in the case of motion.

Fig. 1. Consecutive frames in a clip, from which we observe that the visual contents change dramatically.

Fig. 1 shows some consecutive frames in a clip, where we sample 2 frames every second from the video. Due to the object motion, the visual representations vary greatly for similar content, which makes them act different roles in the short-term context. Therefore, the attention mechanism that assigns different weights to different frames is essential to make them more discriminative. To this end, we propose an encoder self-attention mechanism to alleviate this short-term contextual attention insufficiency issue. To the best of our knowledge, there has been little previous work employing self-attention in video summarization. Combining the proposed self-attention for short-term contextual information and the decoder attention for long-term contextual information [3] together, we constitute a deep attentive model in our proposed video summarization approach.

In addition, both vsLSTM [2] and AVS [3] train their networks with mean square error (mse) loss function, which is weak in reflecting the distribution relation. It is easily observed that the importance score distributions between the predicted one and the ground truth vary greatly even they have a small mse loss, which leads to a suboptimal solution. Fig. 2 lists five possible predicted importance score distributions with the same ground truth and mse value. It can be observed that these predicted sequences differ greatly from the ground truth except for Fig. 2(a) and (b). This demonstrates that mse, as a value-based loss function, is not a good choice for Seq2Seq learning-based video summarization since it cannot guarantee to reflect the distribution of the ground truth. To alleviate this distribution inconsistency issue, we introduce a distribution-based function as a complement for mse. In this way, both the minimal distance and distribution consistency are satisfied for the predicted importance score sequence against the ground-truth one.

The contributions of this article are summarized as follows.

1) Two critical issues in video summarization are discovered, i.e., short-term contextual attention insufficiency and distribution inconsistency. The former refers to that current approaches are deficient in capturing the short-term contextual attention information within the video, which is an important knowledge to be modeled in video summarization. The latter is that the distributions of predicted importance score sequence and the ground-truth sequence are inconsistent, which may lead to a suboptimal solution.
2) It proposes an encoder self-attention mechanism for Seq2Seq learning-based video summarization. It assigns weights to the encoder outputs according to their importance to the short-term context, which is complementary to the encoder–decoder attention mechanism. By doing so, both the short- and long-term contextual attentions are satisfied, which is a prime requirement for effective video summarization.

Fig. 2. Distributions of different predictions, where (a)–(e) have the same ground-truth sequences but different predicted importance score sequences. Although they have different distributions, they have the same mse (mse = 1.0). (a) Possible distribution 1. (b) Possible distribution 2. (c) Possible distribution 3. (d) Possible distribution 4. (e) Possible distribution 5.

3) It presents a distribution-based loss function to overcome the limitation of employing mse individually. On the premise of minimal distance between the sequences of predicted importance score and ground truth, it learns the distribution consistency from them, which guarantees a better usage of the human annotations.

We evaluate our approach via extensive experiments on the benchmark data sets of SumMe [12] and TVSum [13], on which the results outperform the state-of-the-art approaches by a large margin. In addition, we also conduct an ablation study to prove the contribution of our encoder self-attention and distribution consistency loss function.

The rest of this article is organized as follows. Section II reviews the related video summarization methods. Section III introduces the proposed Attentive and Distribution consistent video Summarization (ADSum) approach. Section IV presents the experimental results and analyses. Finally, conclusions and future work are provided in Section V.

II. Related Work

Video summarization, also called video abstraction or video skim, has been an active research topic for more than two decades [3], [14], [15]. Similar works also include video keyframe selection [16], video highlight detection [17], and video story segmentation [18]. The goals for these lines of studies are similar, i.e., to provide a succinct yet informative video subsets (keyframes or keyshots) to facilitate the browsing and understanding of individual videos. It should be noted that this article is concerned with single video summarization but not multiple video summarization [19],

[20], whose purpose is to condense a set of query-related videos into a compact summary. In the following, we first briefly review the conventional approaches and then introduce the state-of-the-art Seq2Seq learning-based approaches.

A. Conventional Approaches

Conventional approaches characterize themselves with handcrafted features and shallow structures. Many of them rely on the unsupervised approaches, such as clustering and sparse coding. Specifically, the clustering-based approaches generally select cluster centers as summary subsets. By doing so, the redundant content can be removed and the video is shortened. Accordingly, many efforts are devoted by employing and designing various clustering methods, ranging from k-means [21], Delaunay clustering [22], graph clustering [23], and prototype selection [4], to archetypal analysis [13]. The sparse coding-based approaches formulate video summarization as a minimum sparse reconstruction problem [24], [25] since the sparsity and reconstruction error terms in it naturally accord with the problem of summarization. For example, Cong *et al.* [24] exploited a sparse coding model to leverage the dictionary as keyframes as they could reconstruct the original video. Instead of using the $L_{2,1}$ norm in [24], Mei *et al.* [25] utilized the L_0 norm in their proposed method. For efficiency consideration, Zhao and Xing [26] exploited a quasi-real-time approach to summarize videos. In addition, motion state change detection is also employed in video summarization. For example, Zhang *et al.* [27] proposed to employ the spatiotemporal slices to analyze the object motion trajectories and select motion state changes as a metric to summarize videos. Specifically, the motion state changes are formulated as a collinear segment on a spatiotemporal slice problem, by which an attention curve is formed to generate the summary.

Besides unsupervised approaches, there has been a dramatic increase in designing supervised learning-based approaches over the past few years. Prior work [1], [7], [28] includes optimizing one or multiple objective functions for video summarization. For example, SVM is employed to classify each segment with importance score in [7], and those segments with higher scores are selected to constitute a video summary. Li *et al.* [1] and Gygli *et al.* [28], respectively, learned a combination function of them with a maximum margin formulation to ensure that the generated summaries are close to the human-created summaries and designed several handcrafted criteria. By resorting to the human annotations, this line of work usually has a better performance than the unsupervised one.

B. Seq2Seq Learning-Based Approaches

Recent years have seen a surge in Seq2Seq learning-based approaches for video summarization with the renaissance of deep learning, especially the LSTM technique [2], [3], [29]. A typical framework for this line of work is to take the original video frame sequence as input and the frame importance score sequence as output and exploits LSTM to capture the long-term contextual information. For example, Zhang *et al.* [2] proposed to utilize BiLSTM as encoder and MLP as decoder in their vsLSTM approach and further introduced determinantal point process (DPP) to vsLSTM to enhance the diversity. To further strengthen the long temporal dependences among video frames, Zhao *et al.* [30] developed a hierarchical architecture of LSTMs by employing an LSTM layer and a BiLSTM layer as encoder and decoder, respectively.

A critical issue of the abovementioned methods is that they considered the whole input frames as equally important, that is to say, all the frames in the input video sequence are treated with the same importance no matter what kind of output frames are to be predicted, which weakens the discrimination of the representative frames. To alleviate this issue, AVS [3] introduces an encoder–decoder attention mechanism in the Seq2Seq framework by conditioning the generative process in the decoder on the encoder hidden states. By doing so, different input frames are assigned different weights, which can provide the inherent relations between the input video sequence and the output keyframes. SASUM [8] proposes to strengthen the semantic attention by resorting to additional text descriptions. It first employs a Seq2Seq model to embed the input visual information into text representation and then presents a frame selector to exploit the embedded text description to find video keyframes that are relevant to the high-level context.

Besides the abovementioned supervised approaches, the Seq2Seq model can also be leveraged in an unsupervised manner [9], [10]. For example, SUM-GAN [9] proposes to formulate video summarization in the framework of generative adversarial network (GAN), where the LSTM is applied as generator and discriminator. Motivated by the success of reinforcement learning, DR-DSN [10] formulates video summarization as a sequential decision-making process, where a reward function judges the qualification degree of the generated summaries, and a Seq2Seq model is encouraged to earn higher rewards by learning to produce more qualified summaries.

Our proposed ADSum formulates video summarization as a supervised Seq2Seq problem. Different from the existing approaches, we exploit additional encoder self-attention to strengthen the short-term contextual attention among input video frames and an effective distribution learning loss function.

III. Proposed Approach

This section introduces our proposed ADSum video summarization approach in detail. As shown in Fig. 3, ADSum formulates video summarization as a Seq2Seq task, in which the input and output are sequences of video frames and their predicted importance scores. Specifically, the video is first downsampled into frame sequence, and GoogleNet is employed to extract the visual features. Next, BiLSTM is selected as an encoder, and the proposed self-attention is applied on it. Then, LSTM is used as a decoder to output the predicted importance scores for each frame, where either additive or multiplicative attention is exploited on it. Finally, the ground truth is employed as supervision against the output importance scores, where both regression loss and the proposed distribution loss are used as the final objective function. These steps constitute the training stage. After the model

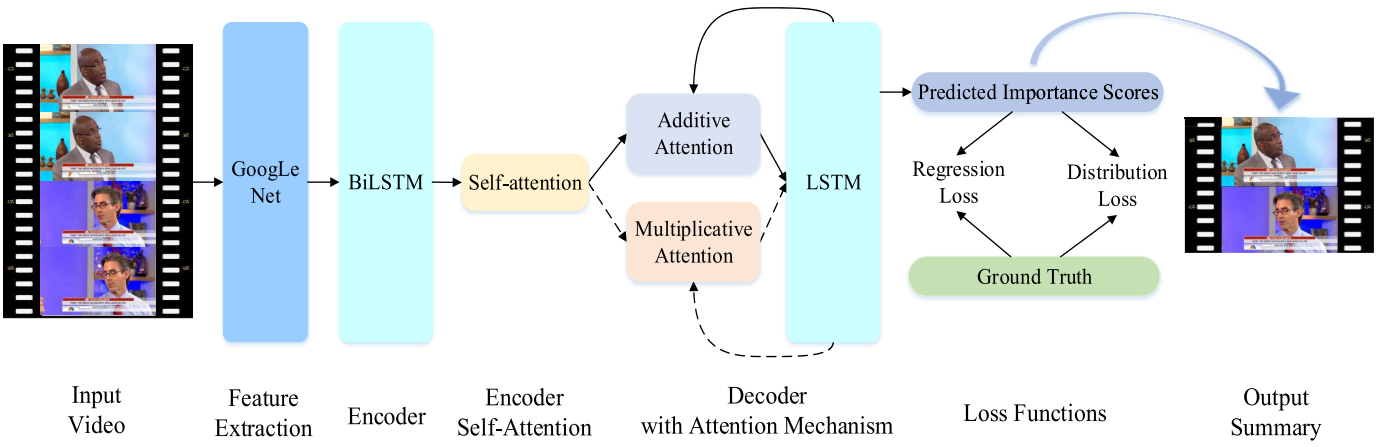


Fig. 3. Illustration of the proposed ADSum approach. The dotted line indicates an alternative step.

training is completed, a predicted importance score sequence will be output when a test video is an input. Then, the summary for the test video is generated according to the output importance score sequence. It can mainly be divided into five components, which are encoder, encoder self-attention, decoder with attention mechanism, loss functions, and summary generation. We introduce these components in detail.

A. Encoder

We first downsample the videos into frame sequences in 2 frames/s. For fair comparison [2], [3], [10], [31], we choose to use the output of pool5 layer of the GoogLeNet [32] (1024 dimensionality), trained on ImageNet [33], as the visual feature for each video frame. Afterward, the features of video frame sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ are fed into a BiLSTM network [34], which is selected as the encoder. BiLSTM is usually employed as the encoder in recent studies of video summarization [3], [8], [35] due to its capability of capturing bidirectional long-term structural dependences among frames. It splits the neurons of a regular LSTM [24] into two directions: one for positive time direction (forward states) and the other for negative time direction (backward states), which are called forward LSTM and backward LSTM. The major difference between forward LSTM and backward LSTM is that the former encodes information from the beginning to the end and the latter encodes information from the end to the beginning. BiLSTM can better capture bidirectional semantic dependences since its output is the concatenation of forward hidden states and backward hidden states, which is shown in Fig. 4.

In forward LSTM, $\mathbf{x}_{t-1}$, $\mathbf{x}_t$, and $\mathbf{x}_{t+1}$ are its inputs, and the corresponding outputs are $\mathbf{h}^f_{t-1}$, $\mathbf{h}^f_t$, and $\mathbf{h}^f_{t+1}$. Meanwhile, the inputs of backward LSTM are $\mathbf{x}_{t+1}$, $\mathbf{x}_t$, and $\mathbf{x}_{t-1}$, and the outputs are $\mathbf{h}^b_{t+1}$, $\mathbf{h}^b_t$, and $\mathbf{h}^b_{t-1}$. The final outputs of BiLSTM $\mathbf{h}$ are the concatenation of forward hidden states and backward hidden states at the same time, i.e., $\mathbf{h}_{t-1} = [\mathbf{h}^f_{t-1}, \mathbf{h}^b_{t-1}]$, $\mathbf{h}_t = [\mathbf{h}^f_t, \mathbf{h}^b_t]$, and $\mathbf{h}_{t+1} = [\mathbf{h}^f_{t+1}, \mathbf{h}^b_{t+1}]$.

B. Encoder Self-Attention

Although BiLSTM is good at capturing the contextual information among video frames, its limitation lies in treating each

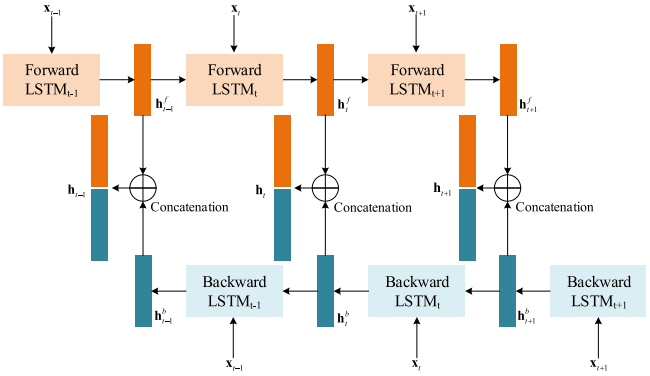


Fig. 4. Flowchart of BiLSTM.

input with equal importance, that is to say, it is insufficient in capturing the attention information within the video sequence itself. To alleviate this issue, we propose to exploit both short- and long-term contextual dependences with a deep attentive mechanism. In this section, we develop to strengthen the short-term contextual dependences with a self-attention mechanism, as shown in Fig. 5.

First, the encoder outputs $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}$ is convolved in one dimension with a convolution kernel of the same size as $\mathbf{h}$, in which the slide step is 1, and it slides T times in total, which generates T weights. The output $\mathbf{H}$ of encoder is fed into a convolutional layer to adjust the encoding representations and obtain original weights belonging to each frame. Next, original weights are mapped into a new space by the sigmoid function to restrict their values between 0 and 1. The formulation is shown as

$$S(\mathbf{H}) = \frac{1}{1 + \exp[\text{conv}(\mathbf{H})]} \tag{1}$$

where conv() denotes a convolutional operation, exp() represents the exponential function, $S(\mathbf{H}) = \{s_1, s_2, \ldots, s_m, \ldots, s_T\}$ means mapping output vectors of original weights, and the attention scale T is the length of $S(\mathbf{H})$, which is set to 9 in this article. Then, the mapping values s_m are fed into the softmax function to obtain the

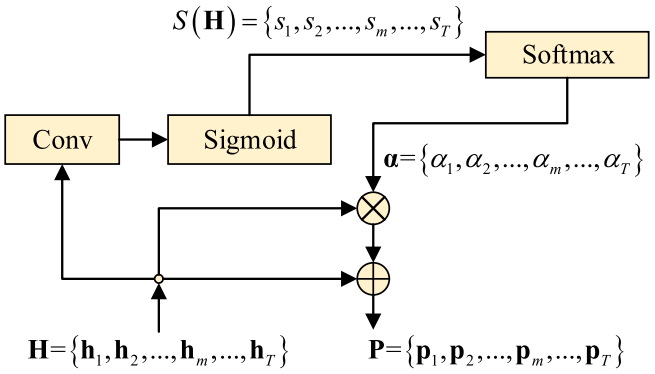


Fig. 5. Flowchart of the proposed encoder self-attention mechanism.

attention weights, which is formulated as

$$\alpha_m = \frac{\exp(s_m)}{\sum_m \exp(s_m)} \tag{2}$$

where α_m represents the weight of the mth frame vector, and it constructs interframe weight vector $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_m, \ldots, \alpha_T\}$. In this way, the encoder self-attention generates and delivers attention weights to each frame. Then, we multiply each weight to its corresponding frame vector to strengthen the differences among them in the same sequence. The attentive encoding representation $\mathbf{p}_m$ is formulated as

$$\mathbf{p}_m = \alpha_m \mathbf{h}_m + \mathbf{h}_m. \tag{3}$$

Since the attention is guided by the encoder outputs $\mathbf{H}$ itself within an attention scale of T, it is a short-term contextual self-attention.

C. Decoder With Attention Mechanism

The decoder is to generate the output importance score sequence $\mathbf{Y} = \{\mathbf{y}'_1, \mathbf{y}'_2, \ldots, \mathbf{y}'_n\}$ conditioned on the sequence features $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_m, \ldots, \mathbf{p}_T\}$ got from encoder. According to the type of output, different decoders can be chosen. For example, if the output is an image or image segments, the CNN is usually chosen as the decoder. When the output is a language or importance score sequence, the LSTM is a better selection. Thus, we employ LSTM as the decoder in our approach. In addition, as discussed in Section II, attention mechanism is capable of modeling of long-term contextual dependences across the entire video, which has been an integrant component in several Seq2Seq-based video summarization approaches [3], [8]. To this end, we employ the attention-based decoder method in [3] in our approach.

Specifically, we can write an LSTM as $f(\mathbf{u}_{n-1}, \mathbf{y}'_{n-1}, \mathbf{p}_m)$, where $f()$ represents LSTM and $\mathbf{u}_n$ is hidden state at time n. Since $\mathbf{p}_m$ is a fixed length encoding vector and cannot accurately reflect the temporal ordering across a long-term video sequence, we apply the attention mechanism to $\mathbf{p}_m$, then we have $\mathbf{q}_n = \sum_{m=1}^{T} \beta_m^n \mathbf{p}_m$, where $\mathbf{q}_n$ represents the attentive vector of $\mathbf{p}_m$, and $\beta_m^n = (\exp(z_m^n))/(\sum_{m=1}^{T} \exp(z_m^n))$ is the mth decoder input attention weight at time n whose sum is 1. In detail, $z_m^n = \phi(\mathbf{u}_{n-1}, \mathbf{p}_m)$ is the similarity score between the hidden states at time $n-1$ and the mth decoder input, where ϕ represents the attention approach. In this way, the decoder is allowed to selectively focus on only a subset of inputs by increasing their attention weights. The attentive decoder is then formulated as

$$\left[P\left(\mathbf{y}'_n | \{\mathbf{y}'_1, \ldots, \mathbf{y}'_{n-1}\}, \mathbf{q}_n\right), \mathbf{u}_n\right] = f\left(\mathbf{u}_{n-1}, \mathbf{y}'_{n-1}, \mathbf{q}_n\right). \tag{4}$$

Then, the key is to choose the attention approach ϕ. According to the relations between the input and the hidden states, similar to [3], we employ either the additive attention approach [36] [see (5)] or multiplicative attention approach [37] [see (6)] to formulize ϕ. In particular, the additive attention approach is written as

$$z_m^n = (\mathbf{W}_3)^T \tanh(\mathbf{W}_1 \mathbf{u}_{n-1} + \mathbf{W}_2 \mathbf{p}_m + \mathbf{b}) \tag{5}$$

and the multiplicative attention approach is formulated as

$$z_m^n = (\mathbf{p}_m)^T \mathbf{W}_4 \mathbf{u}_{n-1} \tag{6}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, and $\mathbf{W}_4$ denote the fully connected layer matrix and $\mathbf{b}$ is the bias. In the abovementioned decoding process, the additive and multiplicative attention models can, respectively, act on the hidden layers of the decoder to obtain different attention vectors of the context. We could observe that the additive attention approach concatenates the video frames and the hidden states of the decoder, whereas the multiplicative attention approach multiplies them to model the relationship. Specifically, the additive attention model adopts a feedforward neural network with a hidden layer to allocate attention. In contrast, the multiplicative attention model employs the matrix operation to assign attention weight, which is more efficient. Accordingly, we can obtain attentive frame-level importance score at the output of decoder. Since the attention is guided by the decoder hidden state $\mathbf{u}_{n-1}$ that contains previous information, it is a long-term contextual attention.

D. Loss Function

The design of loss function plays a critical role in supporting a high-quality video summarization. For example, Jung *et al.* [38] developed a variance loss to ensure the model to predict output scores for each keyframe with high discrepancy, which is simply defined as a reciprocal of variance of the predicted scores. To preserve the semantic information in a video summarization to the original video, Zhang *et al.* [29] proposed a retrospective loss by embedding both the summarization and original video into a shared space and minimizing their distances. Wei *et al.* [8] applied two variants of sparsity losses to force the model to generate satisfying summarizations. Among these loss functions, the regression loss, especially mse, is the most popular loss employed in video summarization [2], [3], [10], [29], and it minimizes the discriminative losses by measuring elementwise discrepancy.

We also apply mse as our loss function, which is the sum of squared distances between the predicted importance scores sequences $\mathbf{Y}'_t$ and ground-truth sequences $\mathbf{Y}_t$ in the tth batch. It is formulated as

$$L_{\text{mse}} = \frac{1}{n} \sum \left(\mathbf{Y}'_t - \mathbf{Y}_t\right)^2 \tag{7}$$

where n is the number of input samples in a batch. However, we could observe that there may exist a lot of possible sequence forms of the predicted importance scores even for the same mse value ($\neq 0$), that is to say, the distributions between the predicted importance scores and the ground truth may vary greatly even they have a small regression loss, as shown in Fig. 2. To better mitigate this distribution inconsistency issue, we propose a distribution consistency learning strategy as a complement for mse loss by employing a simple yet effective Kullback–Leibler (KL) regularization loss term. It aims at seeking a consistent distribution for the predicted importance scores with the ground truth by regularizing their KL divergence. Specifically, we feed both the predicted importance scores and ground truth into the softmax function to embed them in a shared space, which is formulated as

$$\mathbf{SY}_t = \text{softmax}(\mathbf{Y}_t) \tag{8}$$

$$\mathbf{SY}'_t = \text{softmax}\big(\mathbf{Y}'_t\big) \tag{9}$$

where $\mathbf{SY}_t$ and $\mathbf{SY}'_t$ represent the normalized ground-truth sequences and normalized predicted importance score sequences in the tth batch, respectively. In video summarization, high importance score means the corresponding frame or shot could be chosen as keyframe or keyshot. Since the keyframe or keyshot is used to represent its relevant frames/shots, we could approximatively view the normalized importance score of each frame as its probability distribution. To this end, we could describe the KL loss function as

$$L_{kl} = \frac{1}{n}\sum\big(\mathbf{SY}_t \cdot \log(\mathbf{SY}_t) - \mathbf{SY}_t \cdot \log\big(\mathbf{SY}'_t\big)\big). \tag{10}$$

With the KL loss function, we restrict the distribution consistency between the predicted importance scores and the ground truth, which guarantees a better prediction. Accordingly, the final loss function is made up of two parts: mse loss and KL loss, described as follows:

$$L_{mk} = L_{\text{mse}} + \lambda L_{kl} \tag{11}$$

where λ is a balance parameter. The mse loss measures the distance between the predicted importance scores and the ground truth, whereas the KL loss guarantees their distribution consistency.

E. Summary Generation

We follow [2] and [3] to generate the video summarization in forms of keyshots based on the frame-level importance scores. In particular, due to the lack of ground-truth temporal segmentation in video summarization data sets, we first employ the kernel temporal segmentation (KTS) [9] algorithm to split a video into a set of nonintersecting temporal shots. Then, the predicted importance score $\mathbf{Y}'_t$ is divided into shot-level importance scores according to different shots, which are calculated by averaging the frame importance scores within each shot. For the purpose of ensuring that the total summarization length is set to a predefined length ℓ, where we follow [12] to set ℓ to be less than 15% of the length of the original video, we need to solve the following optimization problem:

$$\max \sum_{i=1}^{n} \mu_i \gamma_i, \quad \text{s.t.} \sum_{i=1}^{n} \mu_i \eta_i \leq \ell, \quad \mu_i \in \{0, 1\} \tag{12}$$

where n is the number of shots after video segmentation, and γ_i and η_i are the shot-level importance score and the length of the ith shot, respectively. Notice that $\mu_i \in \{0, 1\}$, where $\mu_i = 1$ indicates that the ith shot is selected as a keyshot. We hope to find the shots of the highest sum of shot-level importance score and ensure the specific video summary length at the same time, which belongs to the 0/1 knapsack problem. After following [13] to solve it with dynamic programming, the keyshots conforming to the summary are obtained. Finally, the summary is created by concatenating those keyshots in a chronological order.

IV. Experiments

A. Experimental Setup

1) Data Sets: We evaluate the proposed ADSum method on three publicly available benchmark data sets of video summarization: SumMe [12], TVSum [13], and YouTube [21]. Specifically, the SumMe data set contains 25 user videos that record a variety of events such as holidays, sports, and history. The videos range from 1.5 to 6.5 min in length. The TVSum data set is a collection of 50 videos from YouTube, which is organized into ten categories, such as grooming an animal, parade, attempting a bike trick, and so on. YouTube has 39 videos, including cartoons, news, and sports. The length of these videos ranges typically from 1 to 10 min. It is worth noticing that there are two types of ground-truth annotations for both TVSum and SumMe data sets: indicator vector (0 or 1) and frame-level importance score vectors. Most approaches [2], [3], [6], [10] employ the frame-level importance scores as the ground truth, and thus, we follow this setting. Since YouTube only provides selected keyframes as ground truths, we set them as our evaluation target directly.

2) Evaluation Metric: We apply the popular F-measure to evaluate the performance of automated generated summary compared with the ground-truth summary [1]–[3], [9], [12], [13], [28]. Similar to [2] and [3], our ADSum approach generates a summary S that is less than 15% in duration of the original. Given a generated summary S and the ground-truth summary G, the precision P and the recall R for each pair of S and G are calculated as a measure with the temporal overlaps between them as follows:

$$P = \frac{\text{overlaped duration of } S \text{ and } G}{\text{duration of } S} \times 100\% \tag{13}$$

$$R = \frac{\text{overlaped duration of } S \text{ and } G}{\text{duration of } G} \times 100\%. \tag{14}$$

Finally, the F-measure is computed as

$$F = \frac{2 \times P \times R}{P + R} \times 100\%. \tag{15}$$

3) Implementation Details: We implement experiments on the Tensorflow platform and all experiments are conducted on an NVIDIA Tesla K40c GPU. We employ BiLSTM with 256 hidden units in an encoder and one-layer LSTM with 256 hidden units in a decoder, and we optimize the network with the gradient descent algorithm and set the learning rate to 0.15. In addition, the batch size is 16 and attention scales are 9. The balance parameter λ in (11) is set to 1. As in the prior work [2], [3], [8], we report the average of F-scores of all testing videos. As for the training/testing data, we apply the same standard supervised learning setting as [2] and [3] where the training and testing are from the disjoint part of the same data set. We employ 20% for testing and the remaining 80% for training and validation, where the proportion of the training set and validation set is also 4:1.

B. Comparison With State-of-the-Art Approaches

Since SumMe and TVSum are most popular data sets, we first perform experiments on them. Then, we provide the experimental result on the YouTube data set.

To demonstrate the superiority of the proposed ADSum approach, 12 state-of-the-art supervised approaches are selected for comparison on SumMe and TVSum data sets, and all of them employ the image CNN as visual features and the experiments are performed in the same settings with ADSum. We retrieve their results from published articles.

Specifically, we are interested in comparing ADSum with those approaches within the Seq2Seq learning-based framework, i.e., vsLSTM [2], dppLSTM [2], SUM-GAN$_{\text{sup}}$ [9], DR-DSN$_{\text{sup}}$ [10], SASUM$_{\text{sup}}$ [8], A-AVS [3], and M-AVS [3]. Concretely, as an early encoder–decoder-based approach, vsLSTM [2] formulates video summarization as a structure prediction problem on sequential data by employing LSTMs for sequence modeling. As a variant for vsLSTM, dppLSTM [2] boosts the diversity by combining LSTM and DPP. By introducing the idea of GAN into the encoder–decoder framework, SUM-GAN$_{\text{sup}}$ [9] makes itself a supervised method by further adding a sparse regularization with the ground-truth summarization labels. Similarly, DR-DSN$_{\text{sup}}$ [10] incorporates the idea of reinforcement learning in the framework, where the reward function accounts for diversity and representativeness. The remaining three approaches are closer to our ADSum since they also exploit the attention mechanism. SASUM$_{\text{sup}}$ [8] presents a semantic attention network by leveraging additional text descriptions as the semantic guidance. The video summarization is implemented by minimizing the distance between the generated text description of the summarized video and the ground-truth text description of the original video and the importance scores between the generated keyframes and the ground truth. AVS [3] is a baseline for our ADSum. It explores the encoder–decoder attention mechanism to assign importance weights to different frames and takes the mse as its loss function. The additive and multiplicative attention mechanisms are leveraged, which corresponds to the approaches of A-AVS and M-AVS.

We also choose five additional supervised approaches without using Seq2Seq learning-based framework for comparison.

TABLE I

F-SCORE (%) PERFORMANCE COMPARISON WITH STATE OF THE ARTS ON THE SUMME AND TVSUM DATA SETS. THE FIRST SECTION AND THE SECOND SECTION SHOW RESULTS WITHOUT AND WITH USING SEQ2SEQ FRAMEWORK. AVERAGE DENOTES THE AVERAGE PERFORMANCE ON BOTH DATA SETS

Method	Feature	SumMe	TVSum	Average
Gygli *et al.*, 2015, [28]	DeCAF	39.7	-	-
Zhang *et al.*, 2016, [39]	AlexNet	40.9	-	-
Li *et al.*, 2017, [1]	VGGNet	43.1	52.7	47.9
HSA-RNN, 2018, [40]	VGGNet	44.1	59.8	52.0
DySeqDPP, 2018, [31]	GoogleNet	44.3	58.4	51.4
vsLSTM, 2016, [2]	GoogleNet	37.6	54.2	45.9
dppLSTM, 2016, [2]	GoogleNet	38.6	54.7	46.7
SUM-GAN$_{\text{sup}}$, 2017, [9]	GoogleNet	41.7	56.3	49.0
DR-DSN$_{\text{sup}}$, 2018, [10]	GoogleNet	42.1	58.1	50.1
SASUM$_{\text{sup}}$, 2018, [8]	GoogleNet	45.3	58.2	51.8
A-AVS, 2019, [3]	GoogleNet	43.9	59.4	51.7
M-AVS, 2019, [3]	GoogleNet	44.4	61.0	52.7
ADSum-A (Ours)	GoogleNet	45.9	**64.5**	**55.2**
ADSum-M (Ours)	GoogleNet	**46.1**	64.3	**55.2**

In particular, the approach of Gygli *et al.* [28] formulated video summarization as a subset selection problem by learning submodular mixtures of objectives for different criteria directly. Zhang *et al.* [39] proposed to learn to transfer summary structures from training videos to test ones. Li *et al.* [1] developed a general framework for both edited and raw videos with the idea of property-weight learning. Considering the influence of video structure on summarization results, HSA-RNN [40] integrates shot segmentation and video summarization into a hierarchical structure-adaptive RNN to jointly exploit the video structure and content. Also, DySeqDPP [31] is a dynamic sequential DPP approach, which aims at enforcing the local diversity in a reinforcement learning manner.

Table I summarizes the comparison results of F-score on the SumMe and TVSum data sets. We can observe that the proposed ADSum achieves the best performance on both data sets. Specifically, on the SumMe data set, ADSum-A and ADSum-M outperform the runner-up approach, SASUM$_{\text{sup}}$, in 0.6% and 0.8%, respectively,and on the TVSum data set, they outperform the runner-up approach, M-AVS, in 3.5% and 3.3%, respectively. Considering the average performance on both data sets, ADSum-A and ADSum-M have interestingly the same performance of 55.2%. It is higher than that of the runner-up approach of M-AVS in 2.5%, which is quite a large margin due to the challenge of the data sets. Besides, we could observe that the top five approaches on average metric all employ attention model, which proves the effectiveness of encoding contextual attentive information. Furthermore, the superiority of the proposed ADSum against the M-AVS, A-AVS, and SASUM$_{\text{sup}}$ mainly lies on the fact of exploiting the self-attention and the distribution consistency loss. Finally, we could observe that all approaches perform better on TVSum than SumMe, which is mainly due to their supervised property, that is to say, since the correlations among videos in TVSum are closer than those in SumMe, it is easier to obtain more useful supervision knowledge for supervised approaches from the training of TVSum data set.

Then, we conduct experiments on the YouTube data set, as shown in Table II. Five state-of-the-art approaches are

TABLE II

F-SCORE (%) PERFORMANCE COMPARISON WITH STATE OF THE ARTS ON THE YOUTUBE DATA SET

Method	F-score(%)
Zhang *et al.*, 2016, [39]	61.0
SUM-GAN$_{\text{sup}}$, 2017, [9]	62.5
A-AVS, 2019, [3]	65.8
M-AVS, 2019, [3]	66.2
Fu *et al.*, 2019, [35]	69.7
ADSum-A (Ours)	70.2
ADSum-M (Ours)	**70.5**

chosen for comparison, whose results are retrieved from [35]. Note that all approaches employ the GoogleNet as the visual features. We could observe that our proposed approaches achieve the best performance. Specifically, ADSum-A and ADSum-M outperform the corresponding baseline A-AVS and M-AVS approaches in 4.4% and 4.3%, respectively. In addition, they outperform the second-best approaches, Fu *et al.* [35], in 0.5% and 0.8%, respectively. The experimental results on YouTube further prove the effectiveness of the proposed approaches.

C. Ablation Studies

To further reflect the impacts of encoder self-attention and the KL loss function, we take the AVS approach as the baseline and conduct the ablation experiments by employing encoder self-attention (AVS+SA) and KL loss function (AVS+KL). Note that the proposed ADSum approach could be considered as AVS+SA+KL. As shown in Table III, the utilization of self-attention and KL loss function contributes to the performance improvements for both additive and multiplicative attention versions. Specifically, it can be clearly observed that the performance gains of A-AVS+SA against A-AVS are 4.7% and 0.2%, and the gains of M-AVS+SA against M-AVS are 3% and 0.4%, both on TVSum and SumMe. These results prove that the short-term contextual attention is quite helpful for video summarization. In addition, there are 4.5% and 0.5% gains for A-AVS+KL against A-AVS, and 2.7% and 0.1% for M-AVS+KL against M-AVS, both on TVSum and SumMe. These results prove that the distribution consistency is really useful for video summarization. Interestingly, it is more obvious for the TVSum data set due to the closer associations among the videos. The results prove the effectiveness of both the proposed components. Moreover, we could find that both ADSum-A and ADSum-M further improve the performance, which demonstrates that both components complement each other. These ablation studies validate our motivation that both encoder self-attention and distribution consistency are helpful to Seq2Seq learning-based video summarization.

D. Experiments on Combined Data Set and Augmented Data Set

It is seen from the abovementioned experiments that our proposed encoder self-attention mechanism and the KL loss function contribute greatly to the improvement of the performance. In the interest of verifying the generalizability of our model, we first make further efforts to experiments on the

TABLE III

ABLATION EXPERIMENTS IN F-SCORE (%)

Method	SumMe	TVSum
A-AVS [3]	43.9	59.4
A-AVS +SA	44.1	64.1
A-AVS +KL	44.4	63.9
ADSum-A (Ours)	**45.9**	**64.5**
M-AVS [3]	44.4	61.0
M-AVS+SA	44.8	64.0
M-AVS+KL	44.5	63.7
ADSum-M (Ours)	**46.1**	**64.3**

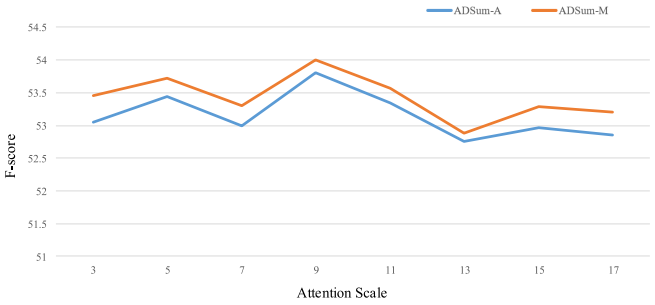


Fig. 6. Parameter sensitivity analysis for attention scales on the Combined data set.

TABLE IV

EXPERIMENTAL RESULTS ON THE COMBINED DATA SET

Method	F-score(%)
A-AVS [3]	52.3
M-AVS [3]	52.7
ADSum-A (Ours)	53.8
ADSum-M (Ours)	**54.0**

combination of both SumMe and TVSum data sets, which is dubbed Combined data set in this article. A similar setting can be found in many existing summarization approaches [2], [40]. Similarly, we also employ 20% for testing and the remaining 80% for training and validation.

There are several observations from Table IV. First, both the proposed ADSum-A and ADSum-M achieve better performance than A-AVS and M-AVS. Second, ADSum-A and ADSum-M have a similar performance, which is consistent with the results of AVS and those shown in Table I. It shows that both multiplicative attention and additive attention are competent for capturing the attentive knowledge in our deep attention framework. Finally, the performances on the Combined data set are a little inferior to those on average in Table I for both ADSum-A and ADSum-M. This lies in the fact that there is a distinct data difference in SumMe and TVSum, which cannot provide enough supervised information to each other after merging.

Then, following [2], [3], and [9], we conduct augmentation experiments on the SumMe and TVSum data sets. Besides the YouTube data set, the OVP data set [41] is also employed as the augmented data set, which has 50 videos from various genres (e.g., documentary and educational) and their lengths vary from 1 to 4 min. Under this setting, given the data set of SumMe or TVSum, we randomly select 20% of it for testing and apply the remaining 80% with the other three data sets to form the augmented training data set. For example, when testing the performance on the SumMe data set, its 20% videos

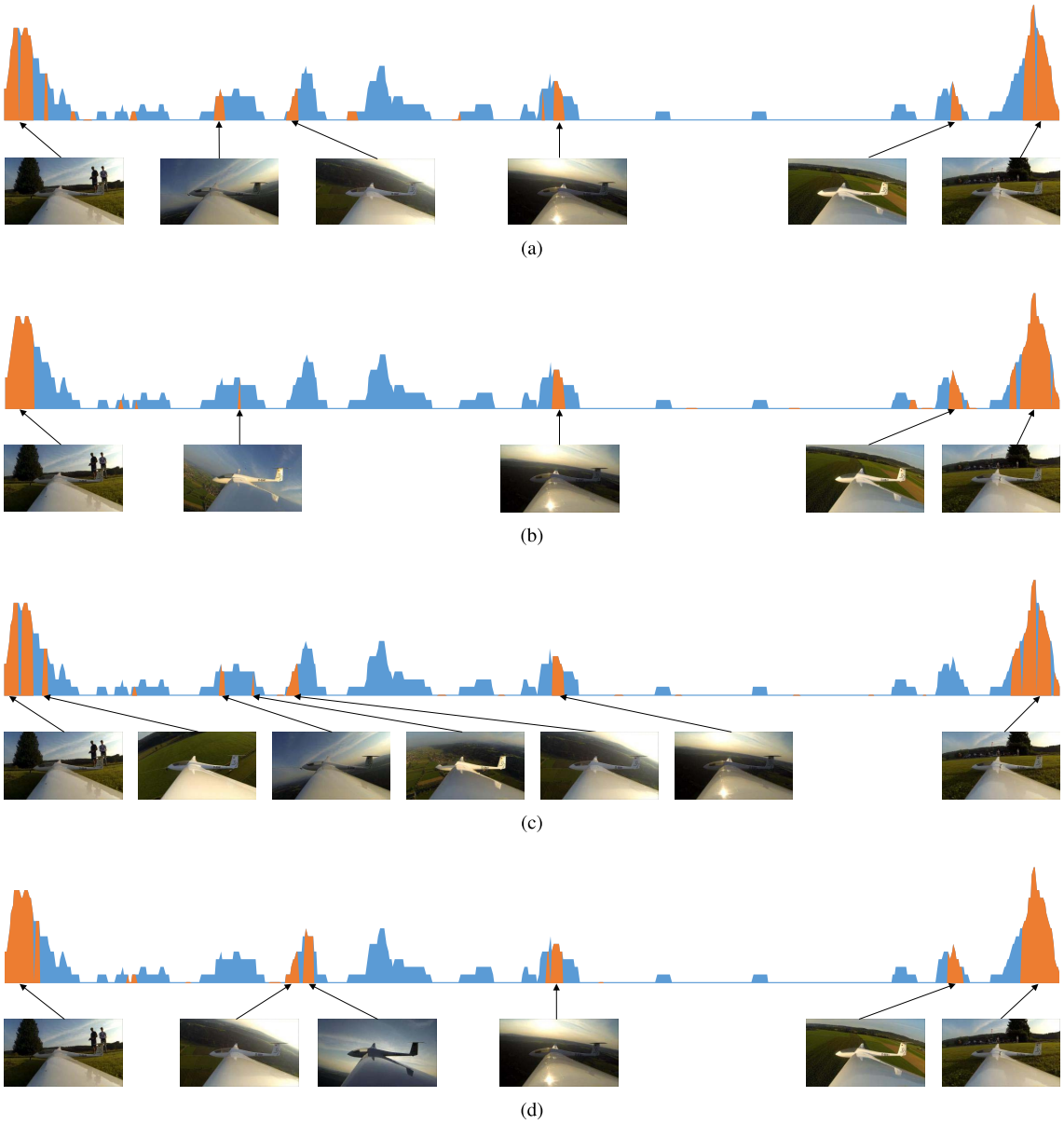


Fig. 7. Exemplar video summaries (orange intervals) from a sample video (Uncut_Evening_Flight of SumMe) along with the ground-truth importance scores (blue background). The corresponding keyframes are ordered in numerical order. (a) A-AVS, F-score (%) = 44.8. (b) M-AVS, F-score (%) = 47.9. (c) ADSum-A, F-score (%) = 50.0. (d) ADSum-M, F-score (%) = 56.2.

are taken for test data, and the remaining 80% videos together with the TVSum, YouTube, and OVP are used as training data set. In this way, the training data sets are augmented. Since more training data are employed, better performance should be obtained by comparing the experimental results shown in Table I, which is referred to as a canonical setting.

Table V shows the results on augmented setting. There are four state-of-the-art approaches are chosen for comparison, which utilizes the same GoogleNet visual features. We could observe that the performances on this setting have consistent improvements against the canonical setting. For example, on the SumMe data set, there are 1.4% and 1.5% improvements for augmented setting against canonical setting for ADSum-A and ADSum-M, respectively. There are also 1.3% and 1.4% improvements on the TVSum data set for two proposed approaches, respectively. This confirms our conjecture that the augmented training data are helpful to improve the generalization of supervised learning approaches. In addition, we could observe that the proposed ADSum-M outperforms the second-best M-AVS approach in 1.5% on the SumMe data set and 3.9% on the TVSum data set, which proves the superiority of our proposed approaches.

E. Parameter Sensitivity Analysis

In this section, we evaluate the impact of attention scale T on the Combined data set. As shown in Fig. 6, when the attention scale value is within the range from 3 to 17, the F-scores of ADSum-A and ADSum-M fluctuate within the range of 1.05% and 1.22%. In addition, it is obvious

TABLE V
EXPERIMENTAL RESULTS ON THE AUGMENTED SETTING

Dataset	Method	Canonical	Augmented
SumMe	dppLSTM [2]	38.6	42.9
	SUM-GAN$_{\text{sup}}$ [9]	41.7	43.6
	A-AVS [3]	43.9	44.6
	M-AVS [3]	44.4	46.1
	ADSum-A (Ours)	45.9	47.3
	ADSum-M (Ours)	**46.1**	**47.6**
TVSum	dppLSTM [2]	54.7	59.6
	SUM-GAN$_{\text{sup}}$ [9]	56.3	61.2
	A-AVS [3]	59.4	60.8
	M-AVS [3]	61.0	61.8
	ADSum-A (Ours)	**64.5**	**65.8**
	ADSum-M (Ours)	64.3	65.7

that the values of ADSum-M under different attention scales are always higher than those of ADSum-A, which indicates the fact that the multiplicative attention model is better at taking advantage of the decoder hidden layer output and the optimized visual features to obtain attentive information. It can be observed that the performance is the best when the attention scale value is 9.

F. Qualitative Results

To get some intuition about qualitative effects on the temporal selection pattern, we visualize some selected keyframes on an example video with a duration of 5.22 min in Fig. 7. It shows the results from A-AVS, M-AVS, ADSum-A, and ADSum-M models on the video "Uncut_Evening_Flight" of SumMe. The video shows a story that some people control a remote-controlled aircraft to shoot an aerial video with a camera attached to the left wing. The blue blocks represent the ground-truth frame-level importance scores, and the marked orange regions are the selected subsets. As shown in Fig. 7, we can observe that the summaries generated by our methods have a more consistent distribution with the ground truth than those generated by AVS models. Besides, the keyframes chosen by our ADSum-A and ADSum-M approaches have larger importance scores than the others.

V. CONCLUSION AND DISCUSSION

In this article, we have proposed a novel deep attentive video summarization approach, called ADSum. It effectively addressed the short-term contextual attention insufficiency and distribution inconsistency issues, which have been neglected before. It considers both the long- and short-term contextual attention with encoder–decoder attention and encoder self-attention in a supervised Seq2Seq framework. In addition, it develops a simple yet effective KL loss for learning the distribution consistency between the predicted importance score sequences and ground-truth sequences. Extensive experiments clearly demonstrate the superiority of ADSum.

One limitation of the ADSum approach is that it is deficient in modeling very long-term contextual attention. This is mainly due to that LSTM is not effective enough in coping with sequence structure longer than 80 time steps [42]. Although we downsample the video into 2 frames/s, 80 time steps are only 40 s. However, one topic in a video may last several minutes. Therefore, it is an important research direction in modeling very long-term important research direction while paying attention to short-term contextual attention.

Another limitation is that it requires large training data. As a supervised learning approach, only sufficient can guarantee a satisfying performance. However, current data sets are still relatively small. One promising way to address this issue is to exploit the idea of transfer learning to transfer the knowledge of other related data sets to video summarization. Of course, a large, well-annotated, publicly available data set is also necessary for promoting the progress of video summarization domain.

In our future work, we will explore some other loss functions to better mimic the way of summarizing videos of human, such as maximum mean discrepancy (MMD) and Wasserstein distance.

REFERENCES

[1] X. Li, B. Zhao, and X. Lu, "A general framework for edited video and raw video summarization," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3652–3664, Aug. 2017.
[2] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 766–782.
[3] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 14, 2019, doi: 10.1109/TCSVT.2019.2904996.
[4] X. Zhang, Z. Zhu, Y. Zhao, D. Chang, and J. Liu, "Seeing all from a few: ℓ_1-norm-induced discriminative prototype selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1954–1966, Jul. 2019.
[5] L. Wu, Y. Wang, L. Shao, and M. Wang, "3-D PersonVLAD: Learning deep global representations for video-based person reidentification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3347–3359, Nov. 2019.
[6] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2069–2077.
[7] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 540–555.
[8] H. Wei, B. Ni, Y. Yan, H. Yu, X. Yang, and C. Yao, "Video summarization via semantic attended networks," in *Proc. AAAI Conf. Art. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 216–223.
[9] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2982–2991.
[10] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 7582–7589.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 505–520.
[13] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5179–5187.
[14] C. Toklu, S.-P. Liou, and M. Das, "Videoabstract: A hybrid approach to generate semantically meaningful video summaries," in *Proc. IEEE Int. Conf. Multimedia Expo*, New York, NY, USA, Jul. 2000, pp. 1333–1336.
[15] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," HP Lab., Palo Alto, CA, USA, Tech. Rep. HPL-2001-191, 2001.
[16] W. Wolf, "Key frame selection by motion analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, USA, May 1996, pp. 1228–1231.
[17] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 982–990.

[18] X. Gao and X. Tang, "Unsupervised video-shot segmentation and model-free anchorperson detection for news video story parsing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 765–776, Sep. 2002.

[19] Z. Ji, Y. Ma, Y. Pang, and X. Li, "Query-aware sparse coding for web multi-video summarization," *Inf. Sci.*, vol. 478, pp. 152–166, Apr. 2019.

[20] Z. Ji, Y. Zhang, Y. Pang, and X. Li, "Hypergraph dominant set based multi-video summarization," *Signal Process.*, vol. 148, pp. 114–123, Jul. 2018.

[21] S. E. F. de Avila, A. P. B. Lopes, A. da Luz, Jr., and A. de Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognit. Lett.*, vol. 22, no. 1, pp. 56–68, 2011.

[22] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *Int. J. Digit. Libraries*, vol. 6, no. 2, pp. 219–232, Apr. 2006.

[23] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Automatic video summarization by graph modeling," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 104–109.

[24] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 66–75, Feb. 2012.

[25] S. Mei, G. Guan, Z. Wang, S. Wan, M. He, and D. D. Feng, "Video summarization via minimum sparse reconstruction," *Pattern Recognit.*, vol. 48, no. 2, pp. 522–533, Feb. 2015.

[26] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2513–2520.

[27] Y. Zhang, R. Tao, and Y. Wang, "Motion-state-adaptive video summarization via spatiotemporal analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1340–1352, Jun. 2017.

[28] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3090–3098.

[29] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 383–399.

[30] B. Zhao, X. Li, and X. Lu, "Hierarchical recurrent neural network for video summarization," in *Proc. ACM Multimedia Conf. (MM)*, Mountain View, CA, USA, Oct. 2017, pp. 863–871.

[31] Y. Li, L. Wang, T. Yang, and B. Gong, "How local is the local diversity? Reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 151–167.

[32] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[33] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[35] T.-J. Fu, S.-H. Tai, and H.-T. Chen, "Attentive and adversarial learning for video summarization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa Village, HI, USA, Jan. 2019, pp. 1579–1587.

[36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.

[37] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, Sep. 2015, pp. 1412–1421.

[38] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan. 2019, pp. 8537–8544.

[39] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1059–1067.

[40] B. Zhao, X. Li, and X. Lu, "HSA-RNN: Hierarchical structure-adaptive RNN for video summarization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7405–7414.

[41] *Open Video Project*. Accessed: May 6, 2017. [Online]. Available: http://www.open-video.org/

[42] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—Video to text," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 4534–4542.

Zhong Ji (Member, IEEE) received the Ph.D. degree in signal and information processing from Tianjin University, Tianjin, China, in 2008.

He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. His current research interests include machine learning, computer vision, multimedia understanding, and video summarization.

Yuxiao Zhao received the B.S. degree in communication engineering from Weifang University, Weifang, Shandong, China, in 2017. She is currently pursuing the master's degree with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China.

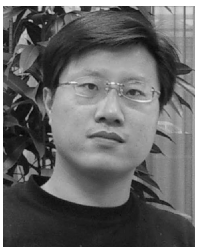
Her research interests include computer vision and video summarization.

Yanwei Pang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2004.

He is currently a Professor with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. He has authored over 80 scientific articles. His current research interests include object detection and recognition, vision in bad weather, and computer vision.

Xi Li received the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China, in 2009.

He was a Senior Researcher with The University of Adelaide, Adelaide, SA, Australia. From 2009 to 2010, he was a Post-Doctoral Researcher with CNRS–Téé com ParisTech, Paris, France. He is currently a Full Professor with Zhejiang University, Zhejiang, China. His current research interests include visual tracking, motion analysis, face recognition, Web data mining, and image and video retrieval.



Jungong Han received the Ph.D. degree in telecommunication and information system from Xidian University, Xi'an, Shaanxi, China, in 2004.

He was a Senior Lecturer with the School of Computing and Communications, Lancaster University, Lancaster, U.K., and the Department of Computer Science, Northumbria University, Newcastle upon Tyne, U.K. He is currently a tenured Associate Professor of data science with the University of Warwick, Warwick, U.K.