Combining Global and Local Attention with Positional Encoding for Video Summarization

Evlampios Apostolidis
CERTH-ITI &
Queen Mary University of London
Thessaloniki, Greece, 57001
Email: apostolid@iti.gr

Georgios Balaouras, Vasileios Mezaris
CERTH-ITI
Thessaloniki, Greece, 57001
Email: {mpalaourg, bmezaris}@iti.gr

Ioannis Patras
Queen Mary University of London
London, UK, E14NS
Email: i.patras@qmul.ac.uk

***Abstract*—This paper presents a new method for supervised video summarization. To overcome drawbacks of existing RNN-based summarization architectures, that relate to the modeling of long-range frames' dependencies and the ability to parallelize the training process, the developed model relies on the use of self-attention mechanisms to estimate the importance of video frames. Contrary to previous attention-based summarization approaches that model the frames' dependencies by observing the entire frame sequence, our method combines global and local multi-head attention mechanisms to discover different modelings of the frames' dependencies at different levels of granularity. Moreover, the utilized attention mechanisms integrate a component that encodes the temporal position of video frames - this is of major importance when producing a video summary. Experiments on two datasets (SumMe and TVSum) demonstrate the effectiveness of the proposed model compared to existing attention-based methods, and its competitiveness against other state-of-the-art supervised summarization approaches. An ablation study that focuses on our main proposed components, namely the use of global and local multi-head attention mechanisms in collaboration with an absolute positional encoding component, shows their relative contributions to the overall summarization performance.**

***Keywords*-video summarization; self-attention; multi-head attention; positional encoding; supervised learning**

I. Introduction

Recent advances in content generation and sharing technologies resulted in video being the most commonly-preferred medium for communication, or to present an event. Humans are more and more engaged with devices integrating video recording and online sharing functionalities (such as smartphones, tablets and wearable cameras). At the same time, social networks (such as Facebook, Instagram, Twitter, TikTok) and video sharing platforms (such as YouTube, Vimeo, Dailymotion) are widely-used as communication means of both amateur and professional users. This technological environment stimulated a tremendous growth of videos over the Web and highlighted the need for technologies that allow users' navigation within endless collections of videos and quick retrieval of the video content that they are looking for. The answer to this demand comes not only from video retrieval technologies but also from technologies for automatic video summarization. The latter allow generating a concise synopsis that conveys the important parts of the full-length video; based on this, viewers can have a quick overview of the whole story without having to watch the entire content.

Several approaches have been proposed to automate video summarization, and methods that are based on deep network architectures represent the current state of the art in the field. A recent study of the literature [1] shows that most approaches utilize RNNs to model the temporal dependence among video frames and learn how to estimate the frames' importance (e.g., [2]–[15]). However, the use of RNNs and their variations (mostly LSTMs [16] and GRUs [17]) for video summarization shows some weaknesses; these relate to the long paths that forward and backward signals have to traverse in the network, which negatively affect the network's ability to model long-range dependencies, and to the limited amount of parallelizable operations during training, as discussed in [5], [18]–[20].

To overcome the aforementioned drawbacks, some works [19], [21] follow a completely different approach. Instead of using RNNs, they model the frames' dependencies using learnable self-attention mechanisms. Nevertheless, these mechanisms i) focus only on the pair-wise similarity of video frames, and ii) are applied to the entire frame sequence. The former observation highlights that the existing self-attention-based summarization approaches ignore the temporal order of the video frames, that is of major importance when creating the video summary. The latter observation points out the risk of performance degradation in the case of long videos, since, as discussed in [19], frames from temporally distant scenes are likely less relevant than the local ones, but the global attention still needs to explore them. This increases the variance of attention values, which negatively impacts the accuracy of the frames' importance estimation.

To tackle the above discussed limitations of the existing self-attention-based summarization methods, we propose a novel supervised approach, called PGL-SUM, which integrates an absolute positional encoding component into a pair of multi-head self-attention mechanisms that model the frames' dependencies at different granularities. Based on the used attention mechanisms the developed network architecture is able to learn how to identify the most important parts of the video both by considering the entire frame sequence and by focusing on smaller parts of it. Building on this knowledge and having integrated a component to model the temporal order of the video frames, the proposed PGL-SUM model is capable of creating concise and temporally coherent video summaries. Our contributions are the following:

- We introduce the use of absolute positional encoding as part of the self-attention mechanism to address the task of video summarization;
- We propose a new architecture that embeds an absolute positional encoding component into global and local multi-head attention mechanisms to learn a better modeling of frames' dependencies from human annotations.

II. Related Work

Various approaches have been introduced to automate video summarization, and the current state of the art is represented by methods utilizing deep network architectures. For the sake of space, in this section we present in brief the relevant literature on supervised video summarization, focusing mostly on approaches utilizing attention mechanisms that are most closely related to the proposed method. For a more complete study of the literature on deep learning and on more conventional approaches, interested readers can refer to [1] and [22], respectively.

One of the first approaches for video summarization was to model the variable-range temporal dependence among frames and learn how to estimate their importance according to ground-truth annotations. For this, some methods utilize structures of RNNs [2]–[5], or Fully Convolutional Sequence Networks [23]. Other works try to tackle issues related to the limited capacity of RNNs and use additional memory either in the form of external storage [9] or by stacking multiple LSTM and memory layers hierarchically [10]. Some algorithms aim to model the evolution of the users' interest by introducing tailored attention mechanisms in classic [6] or sequence-to-sequence [7], [8] RNN-based architectures. Lebron Casas et al. [6] extend the architecture from [2], by embedding an LSTM-based attention layer to either model the frames' temporal dependence and make estimates about their importance, or to form new frame representations for learning how to produce a diverse video summary. Ji et al. [7] formulate video summarization as a sequence-to-sequence learning problem and integrate an attention layer into an LSTM-based encoder-decoder network. This layer gets the encoder's output and the previous hidden state of the decoder and computes a vector with attention values, which subsequently affects the video decoding process. In their following work, Ji et al. [8] introduce an extension of the model from [7], which integrates a semantic preserving embedding network that evaluates the decoder's output with respect to the preservation of the video's semantics using a tailored semantic preserving loss.

Going one step further, a few video summarization methods learn the frames'/fragments' importance by modeling the spatiotemporal structure of the video. For this, a couple of works use convolutional LSTMs in combination with typical CNN-based deep representations [11], or extract more advanced representations with the help of trainable 3D-CNNs [12]. Another approach [24], extracts spatial and temporal information by processing the raw frames and their optical flow maps with CNNs, and learns frames' importance based on a label distribution learning process. Following a different strategy, the method in [13] combines CNNs and GRUs to form spatiotemporal feature vectors that are then used to estimate the level of activity and importance of each frame. Finally, Huang et al. [25] train a neural network for spatiotemporal data extraction and use the extracted information to create a motion curve and identify the video shots. Then, based on human annotations, a self-attention model learns how to estimate the intra-shot importance and select the key-frames/fragments of the video to form a static/dynamic video summary.

Adopting a different strategy to minimizing the distance between the machine-generated and the ground-truth summary, a couple of methods use Generative Adversarial Networks [14], [15]. Their goal is to train a summarizer in order to fool a trainable discriminator when distinguishing the machine- from the user-generated summary.

Last but not least, a few approaches aim to model the frames' dependencies using variants of the self-attention mechanism of the Transformer Network [18]. The first approach to this direction [19], combines a soft self-attention mechanism with a two-layer fully connected network for regression of the frames' importance scores. Liu et al. [21] describe a hierarchical approach which initially defines a set of shot-level candidate key-frames, and then it employs a multi-head attention model to further assess candidates' importance and select the key-frames that form the summary. Li et al. [20] extend the training pipeline of the typical self-attention mechanism, by introducing a processing step that uses the computed attention values and tries to increase the diversity of the visual content of the summary. The estimated attention values (after incorporating information about the frames' diversity) are used to estimate frames' importance and learn summarization from human annotations. Ghauri et al. [26] propose a variation of the architecture from [19], that uses additional representations of the video content. Besides the typical CNN-based features (obtained from pool5 layer of GoogleNet [27] trained on ImageNet), Ghauri et al. use a model of the Inflated 3D ConvNet [28] trained on Kinetics, to extract a set of motion-related features. Each different set of features is fed to a self-attention mechanism and the outputs of these mechanisms are fused to form a common embedding space for representing the video frames.

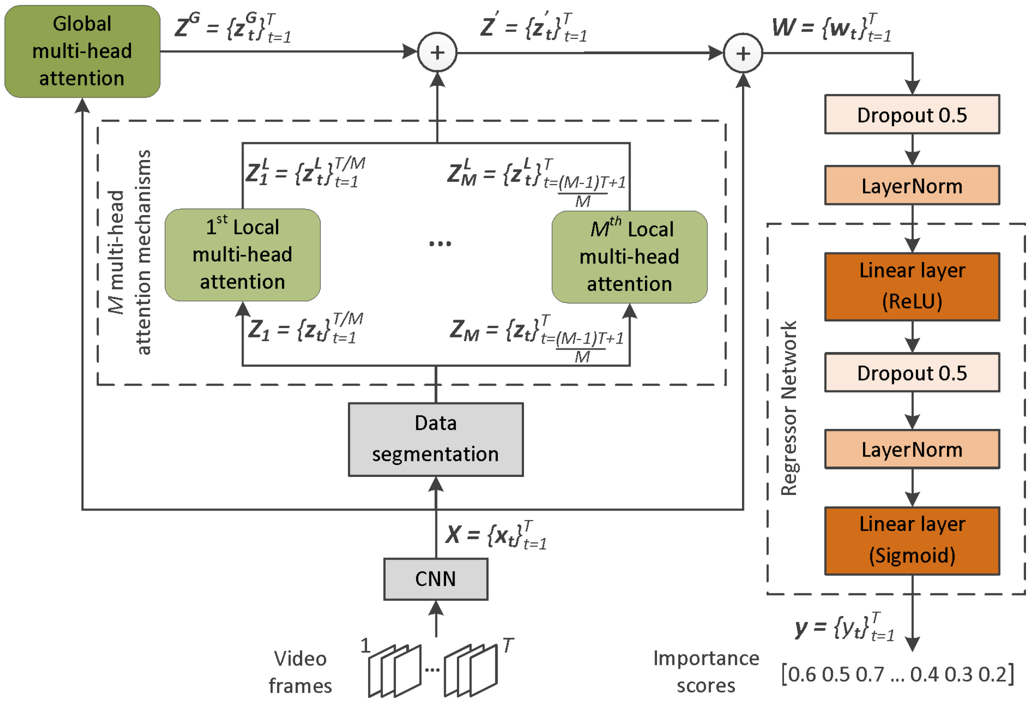
Global multi-head attention
$Z^G = \{z^G_t\}^T_{t=1}$
$Z' = \{z'_t\}^T_{t=1}$
$W = \{w_t\}^T_{t=1}$
Dropout 0.5
LayerNorm
M multi-head attention mechanisms
$Z^L_1 = \{z^L_t\}^{T/M}_{t=1}$
$Z^L_M = \{z^L_t\}^T_{t=\frac{(M-1)T+1}{M}}$
1st Local multi-head attention
...
Mth Local multi-head attention
$Z_1 = \{z_t\}^{T/M}_{t=1}$
$Z_M = \{z_t\}^T_{t=\frac{(M-1)T+1}{M}}$
Data segmentation
$X = \{x_t\}^T_{t=1}$
CNN
Video frames
1
T
Regressor Network
Linear layer (ReLU)
Dropout 0.5
LayerNorm
Linear layer (Sigmoid)
$y = \{y_t\}^T_{t=1}$
Importance scores
[0.6 0.5 0.7 ... 0.4 0.3 0.2]

Figure 1. The PGL-SUM network architecture. Shaded boxes indicate non-trainable parts.

The obtained representation is finally used to learn how to estimate frames' importance.

The proposed approach is most closely related to the methods of the previous paragraph that rely on self-attention mechanisms. Nevertheless, contrary to all these methods that model the frames' dependencies after taking under consideration the entire frame sequence, the developed PGL-SUM model captures the frames' dependence at different granularities with the help of global and local multi-head attention mechanisms. Moreover, in contrast to these methods that completely ignore the sequential nature of the video, our approach enhances each utilized attention mechanism by embedding a component that encodes the temporal position of frames, a type of information that is crucial when estimating the frames' importance. The aforementioned differences target relevant limitations of existing self-attention-based approaches and lead to advanced summarization performance, as reported in Section IV.

III. Proposed Approach

The starting point of our work was the supervised method of Fajtl et al. [19]. The core part of this method is a soft self-attention mechanism that takes the entire frame sequence and models the frames' dependencies based on their pair-wise similarities. The output of this mechanism is then forwarded to a two-layer fully connected network that produces estimates about each frame's importance. These estimates are eventually compared to ground-truth annotations about the frames' importance, thus allowing the network architecture to learn summarization in a supervised manner. This method has two main weaknesses that relate to: i) the complete lack of knowledge about the temporal position of the video frames (which is crucial when producing a temporally-coherent summary of the video content), and ii) the growing difficulty to accurately estimate the frames' importance as the video duration increases.

To overcome these weaknesses we built a new architecture that extends [19] by: i) utilizing a multi-head attention mechanism for modeling the frames' dependencies according to the entire frame sequence, ii) introducing multiple multi-head attention mechanisms that model short-term dependencies via focusing on smaller parts of the video, and iii) enhancing these attention mechanisms with a component that encodes the temporal position of the video frames. In the sequel, we present the developed network architecture (Fig. 1) by describing the processing pipeline from video representation to frames' importance estimation, which is the same during both training and inference. Regarding our notation: capital bold letters denote matrices, small bold letters denote vectors and non-bold letters (either capital or small) denote scalars.

Given a video of T frames, the proposed PGL-SUM model initially produces a set of deep feature representations ($\boldsymbol{X} = \{\boldsymbol{x_t}\}^T_{t=1}$) of size D ($\boldsymbol{x_t} = \{x_{t,i}\}^D_{i=1}$) using a pretrained CNN model. These representations form the input to the trainable part of the architecture and follow two different processing paths. One of these paths includes a

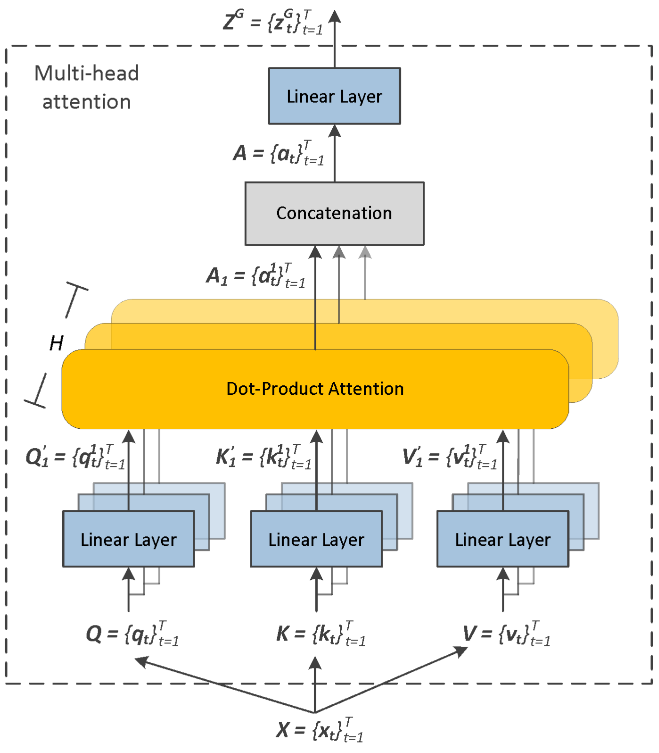

Figure 2. The utilized multi-head attention mechanism.

global multi-head attention mechanism that aims to discover different modelings of the frames' dependencies according to the entire frame sequence. The used multi-head attention mechanism (Fig. 2), which is the same with the one in the Transformer Network [18], gets as input the set of feature vectors ($\boldsymbol{X} = \{\boldsymbol{x_t}\}^T_{t=1}$) and forms the Query ($\boldsymbol{Q} = \{\boldsymbol{q_t}\}^T_{t=1}$), Key ($\boldsymbol{K} = \{\boldsymbol{k_t}\}^T_{t=1}$) and Value ($\boldsymbol{V} = \{\boldsymbol{v_t}\}^T_{t=1}$) matrices of the training process. In the simplest case of single-head attention, these matrices are fed to a triplet of linear layers and the produced embeddings ($\boldsymbol{Q'} = \{\boldsymbol{q'_t}\}^T_{t=1}$, $\boldsymbol{K'} = \{\boldsymbol{k'_t}\}^T_{t=1}$ and $\boldsymbol{V'} = \{\boldsymbol{v'_t}\}^T_{t=1}$ respectively), that maintain the dimensions of $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$, are given as input to the dot-product attention process, which produces the attention values for the video frames ($\boldsymbol{A} = \{\boldsymbol{a_t}\}^T_{t=1}$). In the case of multi-head attention the different heads, that are denoted by the existence of a set of H different linear layers associated to H dot-product attention processes, produce different embeddings for the Query, Key and Value matrices ($\boldsymbol{Q'_j}$, $\boldsymbol{K'_j}$, $\boldsymbol{V'_j}$, with $j \in [1, H]$) in which the dimension of each representation is reduced to D/H. Each triplet of these embeddings (for simplicity, in Fig. 2 we depict the output of only the first triplet of linear layers) is then forwarded to a different dot-product attention process that computes a set of attention values. The computed sets of attention values for the different triplets of embeddings ($\boldsymbol{A_j} = \{\boldsymbol{a^j_t}\}^T_{t=1}$, with $j \in [1, H]$) are then concatenated. The output of this process ($\boldsymbol{A} = \{\boldsymbol{a_t}\}^T_{t=1}$) is fed to a

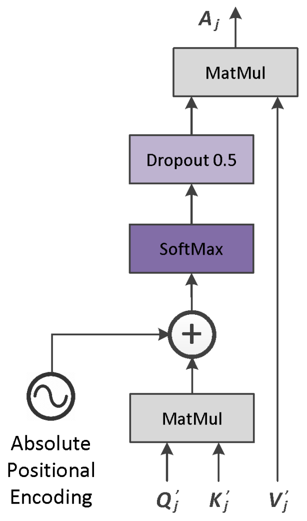

Figure 3. The applied dot-product attention that integrates knowledge about the temporal position of the video frames.

linear layer that forms the final outcome of the global multi-head attention mechanism ($\boldsymbol{Z^G} = \{\boldsymbol{z^G_t}\}^T_{t=1}$). This new representation captures information about each frame's importance according to the entire frame sequence.

Regarding the dot-product attention (Fig. 3), we enhance this process by incorporating knowledge about each frame's position in the frame sequence. For this, we use a component that encodes the absolute position of the frames using sine and cosine functions of different frequencies:

$$\begin{aligned} \mathrm{PE}_{(\mathrm{pos},2i)} &= \sin(\mathrm{pos}/10000^{2i/D}) \\ \mathrm{PE}_{(\mathrm{pos},2i+1)} &= \cos(\mathrm{pos}/10000^{2i/D}) \end{aligned} \tag{1}$$

where pos is the position and i is the index of each frame in the frame sequence. This encoding is similar to the one used in [18]. However, given an item of the frame sequence, we do not apply this encoding over the elements of its representation vector (as in [18]), but over the elements of the sequence. So, given the video of T frames that are represented by D-sized feature vectors, we compute a positional encoding matrix of size $T \times T$ (and not $T \times D$ as in [18]). The encoded information of this $T \times T$ matrix is incorporated into the dot-product attention by being added to the output of a matrix multiplication that involves the Query and the (transposed) Key matrices. A similar idea was described in [29], to model the relative position of the elements of a sentence for addressing the machine translation task. However, this work adopts a threshold about the maximum relative position between the elements of a sequence, that could be quite restrictive in the case of videos. Hence, we prefer to use a component that encodes the absolute position of the video frames. Finally, we avoid the scaling part of the original dot-product attention approach [18], as in our case experiments showed that it negatively affects the method's performance.

The other processing path includes a segmentation step that splits the originally extracted set of deep feature vectors for the video frames ($\boldsymbol{X} = \{\boldsymbol{x_t}\}_{t=1}^{T}$) into M consecutive and non-overlapping segments. Each one of these segments ($\boldsymbol{Z_i}$, with $i \in [1, M]$) contains the deep feature vectors of the video frames that lie within the segment ($\boldsymbol{Z_i} = \{\boldsymbol{z_t}\}_{t=(i-1)\frac{T}{M}+1}^{i\frac{T}{M}}$). Each set of feature vectors is then forwarded to a different local multi-head attention mechanism that focuses on the corresponding part of the video. Based on the analysis pipeline described for the case of the global multi-head attention mechanism, each local attention mechanism produces a new representation of the feature vectors of the frames that lie within the associated segment of the video ($\boldsymbol{Z_i^L}$, with $i \in [1, M]$ and $\boldsymbol{Z_i^L} = \{\boldsymbol{z_t^L}\}_{t=(i-1)\frac{T}{M}+1}^{i\frac{T}{M}}$), that encodes data about the importance of each video frame according to its dependence to the frames within the same segment. The only difference in this processing path is that the produced embeddings by the different heads of each local attention mechanism ($\boldsymbol{Q_j'}$, $\boldsymbol{K_j'}$, $\boldsymbol{V_j'}$, with $j \in [1, H]$) are formed through an additional dimensionality reduction step that relates to the number of video segments and results in representations of dimension equal to $D/(H \cdot M)$. The reason for applying such an additional dimensionality reduction in the case of multi-head local attention mechanisms is to maintain a low level of computational complexity, and it was based on the intuition that smaller representations would be sufficient for modeling dependencies over shorter sequences of video frames. This intuition was experimentally validated after evaluating the performance of a variation of the proposed model that does not perform the additional dimensionality reduction step.

Having available the generated representations from the global ($\boldsymbol{Z^G}$) and the multiple local multi-head attention mechanisms ($\boldsymbol{Z_i^L}$, with $i \in [1, M]$), the next step is to perform feature addition (represented by the $\oplus$ symbol at the left in Fig. 1) and produce a new representation for each video frame, that carries information about each frame's importance according to its global and local dependencies ($\boldsymbol{Z'} = \{\boldsymbol{z_t'}\}_{t=1}^{T}$). The resulting set of representations is then added to the original deep representations ($\boldsymbol{X} = \{\boldsymbol{x_t}\}_{t=1}^{T}$) via a residual skip connection that aims to facilitate back-propagation (this addition is represented by the $\oplus$ symbol at the right in Fig. 1). The output of this operation ($\boldsymbol{W} = \{\boldsymbol{w_t}\}_{t=1}^{T}$) is forwarded to a dropout layer that is followed by a normalization layer. The resulting representation is given as input to the Regressor Network, which is the same with the one in [19]. Finally, the Regressor produces a set of frame-level scores ($\boldsymbol{y} = \{y_t\}_{t=1}^{T}$ with $y_t \in \mathbb{R}$ and $0 \leq y_t \leq 1$) that indicate the frames' importance.

Given the output of the aforementioned processing pipeline, at training time, we compute the Mean Squared Error between this output and the ground-truth annotations (both represent frame-level importance scores). The computed training loss is then back-propagated to compute the gradients and update all the different trainable parts of the architecture. At inference time, the estimated importance scores are used to select the key-fragments of the video and form the video summary. For this, given a temporal segmentation of the video into its building blocks (obtained e.g., using the KTS algorithm [30]) fragment-level importance is calculated by averaging the scores of each fragment's frames. Finally, provided that the summary does not exceed 15% of the video duration (which is a common setting in the relevant literature), we form the video summary by solving the Knapsack problem, similarly to [2], [6]–[9], [11], [13], [14], [19], [20], [23], [24].

IV. Experiments

A. *Datasets and Evaluation Approach*

Datasets. To evaluate the performance of our PGL-SUM model we use two benchmarking datasets. The SumMe dataset [31] contains 25 videos (1-6 min. duration) covering multiple events from both first-person and third-person view. Each video is associated to multiple (15-18) annotations in the form of key-fragments. Moreover, a single ground-truth summary in the form of frame-level importance scores (calculated by averaging the key-fragment user summaries per frame) is also provided for each video, to support supervised training. TVSum is composed of 50 videos (1-11 min. duration) from 10 categories of the TRECVid MED dataset. Each video is annotated by 20 users in the form of frame-level importance scores, and a single ground-truth summary (computed by averaging all users' scores) is available as well.

Evaluation Approach. For fair comparison with the majority of state-of-the-art approaches, we adopt the key-fragment-based evaluation protocol proposed in [2]. The similarity between a machine-generated and a user-defined summary is estimated by computing their overlap using the F-Score (as percentage). So, given a video, we compare the generated summary with the user summaries for this video, and compute an F-Score for each pair of compared summaries. Then, we average the computed F-Scores (for TVSum) or keep the maximum of them (for SumMe) and end up with the final F-Score for this video. The computed F-Scores for the entire set of test videos are averaged to form the final outcome about the method's performance. This protocol is directly applicable on SumMe, as user annotations are in the form of key-fragments. For TVSum, frame-level annotations are converted to key-fragment annotations, following [2], [32]. Finally, we follow the established approach (e.g., [7], [8], [19]) of using 80% of the videos of each dataset for training and the remaining 20% for testing. We run experiments on five different randomly-created splits for each dataset, and in the following we report the average performance over these runs.

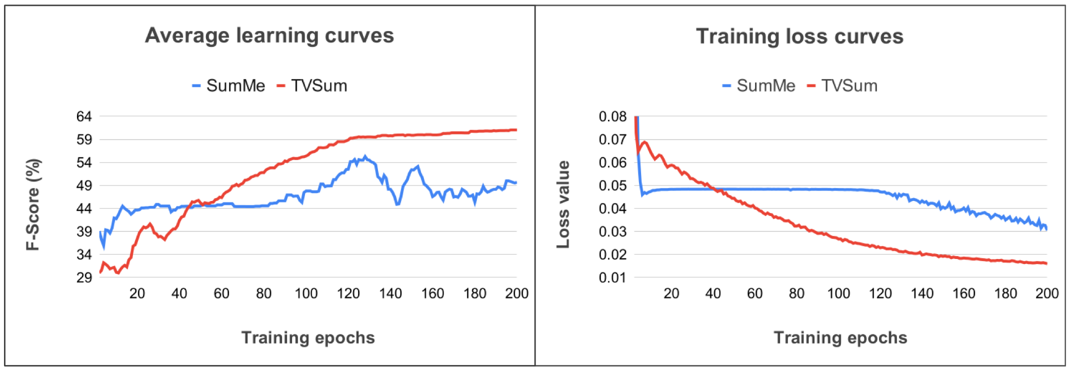

Figure 4. From left to right, the average (over the test sets of the five utilized splits) learning curve and the training loss curve (for one of the used splits) of the proposed method on SumMe and TVSum.

B. Implementation Details

As in most state-of-the-art methods, videos are downsampled to 2 fps. Deep representations of frames are obtained by taking the output of the pool5 layer of GoogleNet [27] trained on ImageNet, and thus are of size $D = 1024$. The number of video segments M, which also determines the amount of local attention mechanisms, equals to 4. The number of the utilized attention heads H is set to 8 and 4 for the global and local attention mechanisms, respectively. Wherever applied, data fusion is based on addition. The learning rate, dropout rate and L2 regularization factor are equal to $5 \cdot 10^{-5}$, 0.5 and 10^{-5}, respectively. For the network's weights initialization, we use the Xavier uniform initialization approach with gain = $\sqrt{2}$ and biases = 0.1. Training is performed in a full-batch mode (i.e., batch size is equal to the number of samples) using the Adam optimizer, and stops after 200 epochs. The aforementioned configuration of the different hyper-parameters and data fusion components of the architecture is made based on the outcomes of the conducted sensitivity analysis that is reported in Section IV-C.

For selecting a well-trained model after the end of the training, we designed a model selection criterion based on the obtained loss curves. The right part of Fig. 4 shows that the training loss is smoothly decreasing for the TVSum data, resulting on a nice convergence in the average (over the test sets of the five used splits) learning curve (left part of Fig. 4). On the other hand, the training loss for the smaller SumMe dataset indicates a more sensitive learning, susceptible to network's initialization and update over the training epochs, which can also be affected by the employed hardware. According to these remarks, the designed criterion focuses on areas of the loss curve that are placed right before rapid changes in the loss value. If such changes are not observed, then the criterion selects a model based on the minimization of the loss value. So, in the case of the SumMe dataset the adopted criterion selects a model after approx. 120 training epochs, thus making a choice that is aligned with the observed performance peak for this dataset in the left part of Fig. 4. In the case of the TVSum dataset, the criterion focuses at the end of the training process, thus selecting a model from one of the last training epochs that, according to the TVSum-related curve on the left part of Fig. 4, appear to have the highest summarization performance.

All experiments were carried out on a PC with an NVIDIA TITAN Xp GPU. To promote reproducibility of our reportings, the PyTorch implementation of PGL-SUM is publicly-available at: https://github.com/e-apostolidis/PGL-SUM.

C. Sensitivity Analysis

We start our experimentation by assessing the sensitivity of the proposed PGL-SUM model with respect to the main hyper-parameters of the network architecture and the different strategies for data fusion. Our plan is to estimate the maximum learning capacity of the network for each one of the different examined configurations. So, as the use of a model selection criterion can impact the evaluation outcome, in this set of experiments we do not consider such a criterion and the summarization performance is formed by averaging the maximum performance of each considered network configuration on the used data splits. The outcomes of this sensitivity analysis will allow to make decisions about the optimal values for the considered hyper-parameters and to select the best data fusion approach.

Our first experiment aims to investigate the optimal choices about the number of video segments (which determines the amount of the utilized local attention mechanisms as well) and the applied strategy for fusing the generated representations by the global and local attention mechanisms of the architecture. The considered cases and the outcomes of this experiment are presented in Table I. The reported values indicate the use of four video segments in combination with an addition-based data fusion approach, as the best option for both datasets.

Following, we examine different choices about the amount of attention heads for the global and local attention mechanisms of the architecture. The results in Table II show

Table I
THE PERFORMANCE (F-SCORE (%)) OF DIFFERENT CONFIGURATIONS OF OUR NETWORK ARCHITECTURE ON SUMME AND TVSUM, THAT RELATE TO DIFFERENT DESIGN CHOICES REGARDING THE NUMBER OF VIDEO SEGMENTS AND THE APPLIED DATA FUSION STRATEGY.

	SumMe			TVSum		
Fusion \ Segments	2	4	8	2	4	8
Addition	49.8	**55.6**	51.9	61.1	**61.7**	59.5
Average pooling	51.5	54.1	52.5	61.1	59.7	58.7
Max pooling	51.1	53.9	52.2	61.3	59.6	58.5
Multiplication	46.3	52.1	52.5	47.3	47.6	46.9

Table II
THE PERFORMANCE (F-SCORE (%)) OF DIFFERENT CONFIGURATIONS OF OUR NETWORK ARCHITECTURE ON SUMME AND TVSUM, THAT RELATE TO DIFFERENT OPTIONS ABOUT THE NUMBER OF HEADS FOR THE GLOBAL AND LOCAL ATTENTION MECHANISMS OF THE NETWORK.

	SumMe			TVSum		
Global \ Local	2	4	8	2	4	8
2	52.4	52.4	52.8	61.4	**61.6**	61.4
4	54.9	58.5	49.2	60.9	61.3	60.4
8	55.8	**58.8**	57.1	60.3	61.1	60.9
16	56.7	57.7	54.9	61.3	60.9	60.1

that the use of 4-head local attention mechanisms is a good choice for both datasets. Concerning global attention, a large number of heads (equal to eight) favors the summarization performance on SumMe, while a smaller number of heads (equal to two) leads to the best performance on TVSum. To avoid dataset-tailored configurations of the network architecture, we decide to use an 8-head global attention mechanism in combination with the four 4-head local attention mechanisms, as this setting leads to consistently good performance on both datasets.

D. Performance Comparisons

Building on the outcomes of the conducted sensitivity analysis, the best configuration of the PGL-SUM network architecture is subsequently compared with a set of supervised video summarization approaches from the bibliography. Our first set of comparisons is based on the experimental evaluation of two attention-based approaches of the literature with publicly-available implementations, that are among the most closely-related ones to our method. In particular, our method is compared with the VASNet method from [19] and the MSVA method from [26], after evaluating all three methods under the exact same experimental conditions; i.e., using the same data splits, adopting the same batch size for training (that equals to one, according to the publicly-available implementations of the VASNet and MSVA methods), and applying the same evaluation approach that takes the average of the maximum values recorded for the videos of the test set of each utilized data split. The results of this experiment, presented in Table III, show that our method significantly outperforms the other two approaches on the SumMe dataset, while being also slightly better on the TVSum dataset. Moreover, the comparison of our method with VASNet - which was the basis for our developments - indicates the positive impact of the introduced extensions to the network architecture of VASNet.

Table III
COMPARISON WITH DIFFERENT SUPERVISED ATTENTION-BASED SUMMARIZATION APPROACHES WITH PUBLICLY-AVAILABLE IMPLEMENTATIONS, ON SUMME AND TVSUM. F1 DENOTES F-SCORE (%) AND RNK DENOTES THE RANKING OF THE COMPARED METHODS.

	SumMe		TVSum		Avg	Data
	F1	Rnk	F1	Rnk	Rnk	splits
VASNet [19]	50.0	3	62.5	2	2.5	5 Rand
MSVA [26]	54.0	2	62.4	3	2.5	5 Rand
PGL-SUM (Ours)	**57.1**	1	**62.7**	1	**1**	5 Rand

Following, the proposed method is compared with a random summarizer and several state-of-the-art supervised video summarization approaches. In this case, our PGL-SUM network architecture is trained in a full-batch mode, and a well-trained model is selected using the criterion described in Section IV-B. Hence, somewhat lower performance than the one reported in Table III can be recorded. The performance of a random summarizer on a given video is measured by: i) randomly assigning importance scores to the video frames based on a uniform distribution of probabilities, ii) computing fragment-level scores based on a predefined KTS-based segmentation of the video, iii) using the Knapsack algorithm to form a summary with a length that does not exceed 15% of video duration. Random summarization is performed 100 times and we report the average score. The performance of each compared supervised method is from the corresponding paper, as the source code for implementing and evaluating these methods on the used data splits is not publicly-available. The reported values in Table IV show that the proposed approach is the best-performing one compared to other existing attention-based summarization approaches (marked with * in Table IV). Moreover, our method is the second-best among a large set of state-of-the-art supervised summarization methods, performing consistently well in both datasets. Contrarily, the top-performing method on TVSum (MAVS [9]) shows random performance on SumMe. The best-performing method on SumMe (SMN [10]) exhibits very good performance on TVSum as well (being the second-best on this dataset). However, according to [10], this method has been evaluated using only one randomly-created split of the used data. As discussed in [33], these random data splits show significantly varying levels of difficulty that affect the evaluation outcomes. To alleviate this effect to some extent, most

Table IV
COMPARISON WITH SUPERVISED VIDEO SUMMARIZATION APPROACHES ON SUMME AND TVSUM. APPROACHES THAT INTEGRATE ATTENTION MECHANISMS ARE MARKED WITH *. F1 DENOTES F-SCORE (%) AND RNK DENOTES THE RANKING OF THE COMPARED METHODS.

	SumMe		TVSum		Avg	Data
	F1	Rnk	F1	Rnk	Rnk	splits
Random Summary	40.2	19	54.4	16	17.5	-
vsLSTM [2]	37.6	22	54.2	17	19.5	1 Rand
dppLSTM [2]	38.6	21	54.7	15	18	1 Rand
ActionRanking [13]	40.1	20	56.3	14	17	1 Rand
*vsLSTM+Att [6]	43.2	16	-	-	16	1 Rand
*dppLSTM+Att [6]	43.8	15	-	-	15	1 Rand
H-RNN [3]	42.1	17	57.9	12	14.5	-
*A-AVS [7]	43.9	14	59.4	8	11	5 Rand
*SF-CVS [25]	46.0	9	58.0	11	10	-
SUM-FCN [23]	47.5	7	56.8	13	10	M Rand
HSA-RNN [4]	44.1	13	59.8	7	10	-
CRSum [12]	47.3	8	58.0	11	9.5	5 FCV
MAVS [9]	40.3	18	**66.8**	1	9.5	5 FCV
TTH-RNN [5]	44.3	12	60.2	6	9	-
*M-AVS [7]	44.4	11	61.0	4	7.5	5 Rand
SUM-DeepLab [23]	48.8	5	58.4	10	7.5	M Rand
*DASP [8]	45.5	10	63.6	3	6.5	5 Rand
*SUM-GDA [20]	52.8	3	58.9	9	6	5 FCV
SMLD [24]	47.6	6	61.0	4	5	5 FCV
*H-MAN [21]	51.8	4	60.4	5	4.5	5 FCV
SMN [10]	**58.3**	1	64.5	2	**1.5**	1 Rand
PGL-SUM (Ours)	55.6	2	61.0	4	3	5 Rand

works use more than one splits in their evaluations (see the rightmost column of Table IV). Moreover, the SMN model [10] relies on the use of LSTMs, and thus its training can not be made in a fully-parallel way. Despite the fact that our PGL-SUM model has 3 times more learnable parameters than the SMN model, comparisons with a baseline variation of this model (called "Stack-LSTM" in [10]) that contains less parameters than SMN, showed that our model is trained 2.5 and 6 times faster on SumMe and TVSum, respectively.

E. Ablation Study

To evaluate the contribution of each of the core components of our model, we conduct an ablation study that includes the following variants of the proposed architecture:

- **PGL-SUM w/o global attention**. This variant leaves out the global attention mechanism and uses only local attention to model the frames' dependencies.
- **PGL-SUM w/o local attention**. This variant does not contain any local attention mechanism and the modeling of frames' dependencies relies on global attention.
- **PGL-SUM w/o positional encoding**. This variant excludes the absolute positional encoding component that is originally used when computing the dot-product attention.

We run this experiment on the same group of five randomly-created data splits and report the average performance. The results in Table V show that the combination of global and local attention mechanisms is a good approach for modeling frames' dependencies, as the removal of one of these mechanisms (and especially of the global one) severely affects the summarization performance on at least one of the used datasets. Moreover, the positional encoding component is also shown to have a positive impact, as it leads to improved performance on SumMe, while allowing the network to maintain the same high levels of summarization performance on TVSum.

Table V
ABLATION STUDY BASED ON THE PERFORMANCE (F-SCORE (%)) OF THREE VARIANTS OF THE PROPOSED MODEL, ON SUMME AND TVSUM.

	SumMe	TVSum
PGL-SUM w/o global attention	46.9	52.4
PGL-SUM w/o local attention	46.7	59.9
PGL-SUM w/o positional encoding	53.1	**61.0**
PGL-SUM (Proposed)	**55.6**	**61.0**

V. CONCLUSION

In this work, we proposed a new network architecture for supervised video summarization, that aims to overcome limitations of existing approaches with respect to: i) the modeling of long-range frames' dependencies, ii) the parallelization ability of the training process (two drawbacks of existing RNN-based methods), and iii) the granularity level at which the temporal dependencies between frames are modeled (a weakness of existing self-attention-based methods). The developed PGL-SUM model uses a number of multi-head attention mechanisms that integrate a component for encoding the temporal position of the video frames. One of these mechanisms aims to model the temporal dependence of frames according to the entire frame sequence, while the remaining ones try to discover modelings of such temporal dependencies by focusing on smaller parts of the video. An ablation study documented the positive contribution of our proposals, namely the use of global and local multi-head attention mechanisms in combination with an absolute positional encoding component. Experiments on two benchmarking datasets (SumMe and TVSum) showed that our PGL-SUM model is the best-performing one compared to existing methods that rely on self-attention mechanisms, and demonstrated its competitiveness against other state-of-the-art supervised summarization approaches.

ACKNOWLEDGMENT

This work was supported by the EU Horizon 2020 programme under grant agreement H2020-832921 MIRROR, and by EPSRC under grant No. EP/R026424/1.

REFERENCES

[1] E. Apostolidis *et al.*, "Video Summarization Using Deep Neural Networks: A Survey," *Proceedings of the IEEE*, accepted for publication, 2021.

[2] K. Zhang *et al.*, "Video Summarization with Long Short-Term Memory," in *Europ. Conf. on Comp. Vision 2016*. Cham: Springer Int. Publishing, 2016, pp. 766–782.

[3] B. Zhao *et al.*, "Hierarchical Recurrent Neural Network for Video Summarization," in *25th ACM Int. Conf. on Multimedia* NY, USA: ACM, 2017, pp. 863–871.

[4] B. Zhao *et al.*, "HSA-RNN: Hierarchical Structure-Adaptive RNN for Video Summarization," in *2018 IEEE Conf. on Comp. Vision and Pattern Rec.*, pp. 7405–7414.

[5] B. Zhao *et al.*, "TTH-RNN: Tensor-Train Hierarchical Recurrent Neural Network for Video Summarization," *IEEE Trans. on Industrial Electronics*, vol. 68, no. 4, pp. 3629–3637, 2020.

[6] L. Lebron Casas *et al.*, "Video Summarization with LSTM and Deep Attention Models," in *25th Int. Conf. on Multimedia Modeling*. Cham: Springer Int. Publishing, 2019, pp. 67–79.

[7] Z. Ji *et al.*, "Video Summarization With Attention-Based Encoder–Decoder Networks," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 30, no. 6, pp. 1709–1717, 2020.

[8] Z. Ji *et al.*, "Deep Attentive and Semantic Preserving Video Summarization," *Neurocomputing*, vol. 405, pp. 200–207, 2020.

[9] L. Feng *et al.*, "Extractive Video Summarizer with Memory Augmented Neural Networks," in *26th ACM Int. Conf. on Multimedia*. NY, USA: ACM, 2018, pp. 976–983.

[10] J. Wang *et al.*, "Stacked Memory Network for Video Summarization," in *27th ACM Int. Conf. on Multimedia*. NY, USA: ACM, 2019, pp. 836–844.

[11] S. Lal *et al.*, "Online Video Summarization: Predicting Future to Better Summarize Present," in *IEEE Winter Conf. on Applic. of Comp. Vision*. HI, USA: IEEE, 2019, pp. 471–480.

[12] Y. Yuan *et al.*, "Spatiotemporal Modeling for Video Summarization Using Convolutional Recurrent Neural Network," *IEEE Access*, vol. 7, pp. 64 676–64 685, 2019.

[13] M. Elfeki *et al.*, "Video Summarization Via Actionness Ranking," in *IEEE Winter Conf. on Applic. of Comp. Vision* HI, USA: IEEE, 2019, pp. 754–763.

[14] Y. Zhang *et al.*, "DTR-GAN: Dilated Temporal Relational Adversarial Network for Video Summarization," in *ACM Turing Celebration Conf.* NY, USA: ACM, 2019, pp. 89:1–89:6.

[15] T.-J. Fu *et al.*, "Attentive and Adversarial Learning for Video Summarization," in *IEEE Winter Conf. on Applic. of Comp. Vision* HI, USA: IEEE, 2019, pp. 1579–1587.

[16] S. Hochreiter *et al.*, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *2014 Conf. on Empirical Methods in Natural Language Processing*. Doha, Qatar: ACL, 2014, pp. 1724–1734.

[18] A. Vaswani *et al.*, "Attention is All You Need," in *31st Int. Conf. on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[19] J. Fajtl *et al.*, "Summarizing Videos with Attention," in *Asian Conf. on Comp. Vision 2018 Workshops*. Cham: Springer Int. Publishing, 2018, pp. 39–54.

[20] P. Li *et al.*, "Exploring Global Diverse Attention via Pairwise Temporal Relation for Video Summarization," *Pattern Recognition*, vol. 111, no. 107677, 2021.

[21] Y.-T. Liu *et al.*, "Learning Hierarchical Self-Attention for Video Summarization," in *2019 IEEE Int. Conf. on Image Processing*. IEEE, 2019, pp. 3377–3381.

[22] V. V. K., D. Sen, and B. Raman, "Video Skimming: Taxonomy and Comprehensive Survey," *ACM Computing Surveys*, vol. 52, no. 5, 2019.

[23] M. Rochan *et al.*, "Video Summarization Using Fully Convolutional Sequence Networks," in *Europ. Conf. on Comp. Vision 2018*. Cham: Springer Int. Publishing, 2018, pp. 358–374.

[24] W.-T. Chu *et al.*, "Spatiotemporal Modeling and Label Distribution Learning for Video Summarization," in *IEEE 21st Int. Workshop on Multimedia Signal Processing*. IEEE, 2019, pp. 1–6.

[25] C. Huang *et al.*, "A Novel Key-Frames Selection Framework for Comprehensive Video Summarization," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 577–589, 2020.

[26] J. Ghauri *et al.*, "Supervised Video Summarization Via Multiple Feature Sets with Parallel Attention," in *2021 IEEE Int. Conf. on Multimedia and Expo*. CA, USA: IEEE, 2021, pp. 1–6.

[27] C. Szegedy *et al.*, "Going Deeper with Convolutions," in *2015 IEEE Conf. on Comp. Vision and Pattern Rec.*, pp. 1–9.

[28] J. Carreira *et al.*, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *2017 IEEE Conf. on Comp. Vision and Pattern Rec.*, pp. 4724–4733.

[29] P. Shaw *et al.*, "Self-Attention with Relative Position Representations," in *2018 Conf. of the North American Chapter of the ACL: Human Language Technologies, Vol. 2*. LA, USA: ACL, 2018, pp. 464–468.

[30] D. Potapov *et al.*, "Category-Specific Video Summarization," in *Europ. Conf. on Comp. Vision 2014*. Cham: Springer Int. Publishing, 2014, pp. 540–555.

[31] M. Gygli *et al.*, "Creating Summaries from User Videos," in *Europ. Conf. on Comp. Vision 2014*. Cham: Springer Int. Publishing, 2014, pp. 505–520.

[32] Y. Song *et al.*, "TVSum: Summarizing Web Videos Using Titles," in *2015 IEEE Conf. on Comp. Vision and Pattern Rec.*, pp. 5179–5187.

[33] E. Apostolidis *et al.*, "Performance over Random: A Robust Evaluation Protocol for Video Summarization Methods," in *28th ACM Int. Conf. on Multimedia*. NY, USA: ACM, 2020, pp. 1056–1064.