Structural Relationship Learning for Video Summarization via Graph Filtering

Anonymous CVPR 2021 submission

Paper ID 11901

Abstract

Video summarization is a challenging task as global structural relationship of a video needs to be well learned to generate a desirable video summary, which not only presents the whole storyline of a video but also discards redundant information. Most existing methods use recurrent neural networks to model the sequential structures in videos. However, they are typically less effective in capturing long-range dependencies. Graph convolutional networks have been demonstrated to be superior in jointly modeling structural relationship and content information. In this paper, we propose an effective architecture for video SUMmarization by designing a Graph Convolutional Network with a Low-pass Graph Filter, named SUM-GCN-LGF. The GCN-LGF can reduce noises and obtain smoother representations, making the downstream video summarization task easier. A novel loss function is designed to make generated summaries comprised of keyframes which can reconstruct video structure relations instead of video features. The architecture can work in both supervised and unsupervised settings for video summarization. Experimental results on two benchmark datasets validate the effectiveness of the proposed structure preserving method in comparison with state-of-the-art approaches.

1. Introduction

With the rapid development of multimedia technology and the widespread use of social network and online video platforms in recent years, how to efficiently organize, represent, store, manage as well as retrieve and quickly browse such a huge amount of video data to obtain useful information becomes increasingly important. Consequently, automatic video summarization, which aims to remove redundancy in videos without losing important information, is receiving growing interests from both academia and industry.

There have been many studies for video summarization over the past few years. Most early methods were based on unsupervised learning [39, 24, 10, 23, 20, 34, 5, 15, 13], mostly using hand-crafted spatio-temporal features and clustering techniques to select frames closest to cluster centroid as summary subsets. These traditional summarization methods usually treated video frames independently without considering structural relationship of video data. Inspired by the great successes of deep learning for computer vision tasks, some current state-of-the-art methods formalized video summarization as a sequence modeling task, utilizing Long Short-Term Memory (LSTM) networks [36, 19, 11, 35] or fully convolutional sequence networks [29]. These deep learning supervised models can learn universal information from ground truth summaries that are hard to be captured with hand-crafted features. Unsupervised learning architectures were also designed by adopting Generative Adversarial Network (GAN) combined with encoder-decoder framework, as done in [2]. Zhou et al. proposed a reinforcement learning framework to tackle unsupervised video summarization [38]. Some summarization methods attempted to obtain additional cues from web images/videos [3, 25] or unpaired data [28] with weak supervision to improve the performance.

The end-to-end deep learning models are more general and adaptive to different types of video. However, it is known that modeling long-range dependency across hundreds even thousands of video frames is challenging for LSTM, as signals have to traverse a long path to access history information at early time. Moreover, global relations between frames cannot be well captured by sequence modeling. Motivated by the success of attention mechanism in various domains, some methods introduced self-attention to character closeness of semantic connections between pairwise frames [8, 12, 16], which have achieved remarkable performance.

Graph-based methods have been demonstrated to be an efficient way of structural relation reasoning. Different from global self-attention which only focuses on delivering information between pairwise frames, graph-based models have the advantage of directly reasoning on graph structure. Recently, graph convolutional networks (GCN) [14] have shown superiority in various computer vision applica-

* Equal contributions
† Corresponding author

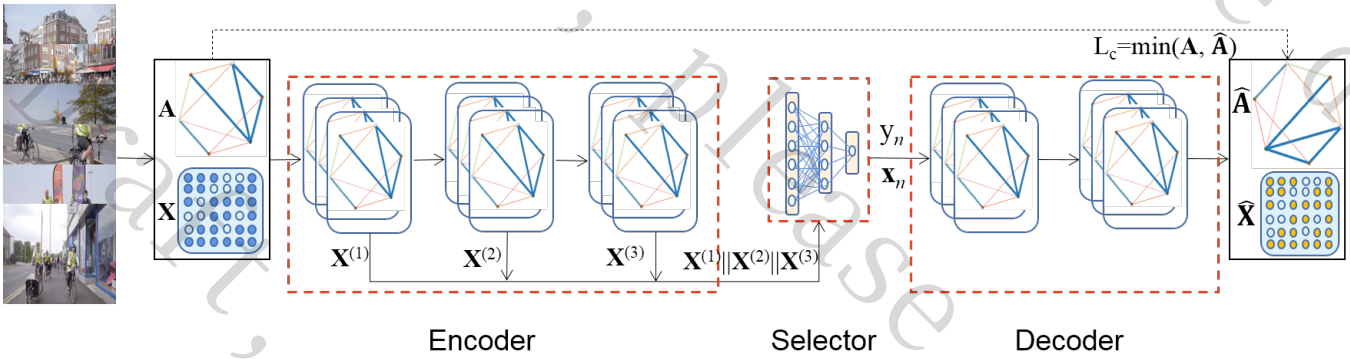


Figure 1. The architecture of SUM-GCN-LGF, consisting of a three-layer GCN-LGF as the encoder, a two-layer fully connected network taking in multiscale features as the selector, and a two-layer GCN-LGF as the decoder. Note that the objective of our decoder is to reconstruct video structures, not video features.

tions, especially in semi-supervised classification. In this paper, we propose a novel framework to learn structural relationship for video summarization using GCN with a Low-pass Graph Filter (GCN-LGF). The proposed architecture is shown in Figure 1. We treat each video frame as a node of a relation graph, following the work of SumGraph [26]. A low-pass graph filter is designed in our GCN to obtain smoother video features. In the encoder network, we combine multiscale features to capture both low-level and high-level semantic information and make better prediction. The selector network is to select keyframes and the decoder network is to reconstruct the video structure. Compared with SumGraph in which the relation graph is estimated recursively, our model is structure-preserving and highly efficient as well as effective.

In summary, our contributions are as follows: (i) a new efficient and effective GCN-based structural relationship learning method is proposed for video summarization. (ii) a low-pass graph filter is exploited in GCN which prompts the network to obtain smoother features. (iii) a new reconstruction loss function is designed to preserve video structure.

2. Related Work

2.1. Video Summarization

Automatic video summarization generates a structured output in the form of a sequence of selected keyframes or clips. It is often modeled as a structured subset selection problem to extract summaries that are most important, representative and diverse.

Most of conventional methods are based on unsupervised learning, such as cluster-based and sparse reconstruction approaches. Cluster-based approaches aggregated frames with high visual similarity into the same cluster. Afterwards they selected frames closest to the cluster centroid from each cluster as the keyframes. Various clustering methods have been adopted, such as graph clustering [24], k-medoid [10], Delaunay clustering [23] and density-based clustering [20, 34]. Sparse reconstruction approaches were mainly based on dictionary learning [21, 22, 7], where a dictionary of key frames was selected such that the original video can be best reconstructed from this representative dictionary.

Deep learning based methods have been successfully applied in video summarization in recent years, especially the LSTM networks. Casting the problem as a structured prediction problem on sequential data, Zhang et al. introduced LSTM to model the variable-range dependencies in the task of video summarization [36]. Mahasseni et al. proposed an unsupervised video summarization method that combines variational recurrent auto-encoders and GAN with LSTM [19]. Then several studies improved the performance by designing different adversarial LSTM architectures, such as attentive conditional GAN [11] and cycle-consistent adversarial LSTM networks [35]. Zhou et al. adopted reinforcement learning to realize unsupervised video summarization [38]. Most of these unsupervised approaches can also be leveraged in supervised manner and obtain better results. Lately methods have achieved more satisfactory performance by using attention mechanism to model the temporal pairwise relation to quantify the importance of each frame. To take the full advantage of GPU parallelization, Rochan et al. proposed fully convolutional models [29] to solve the sequence labeling problem allowing better GPU parallelization than LSTM. Considering that supervised approaches require expensive labeled training data, Rochan et al. also proposed to learn video summarization from unpaired data [28].

Neither sequence modeling networks nor self-attention methods can directly reason based on global structure of video data. Inspired by the success of GCN on relationship reasoning, Park et al. proposed recursive GCN, referred as SumGraph [26], to obtain the optimal relation graph of

an input video and achieved state-of-the-art performance. However, recursively estimating the relation graph in SumGraph resulted in a high computational cost.

2.2. Graph Convolutional Networks

GCN has been widely used in semi-supervised learning, which adopted a localized first-order approximation of spectral graph convolution to avoid the expensive eigen-decomposition [14]. A non-directed graph including N nodes can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of nodes, $\mathcal{E}$ is the set of edges which can be represented by an adjacency matrix $\mathbf{A} = \{a_{ij}\} \in \mathbf{R^{N\times N}}$, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T \in \mathbf{R^{N\times c}}$ is the feature matrix of the nodes, where c is the dimension of node feature. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathrm{diag}(d_1, d_2, ..., d_N)$ is the degree matrix with $d_i = \sum_{j=1}^{N} \mathbf{A}_{ij}$.

The defined layer-wise propagation rule in standard GCN is

$$\mathbf{X}^{(k+1)} = \sigma(\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\mathbf{X}^{(k)}\mathbf{W}^{(k)}) \quad (1)$$

where $\mathbf{X}^{k+1} \in \mathbf{R^{N\times c_{k+1}}}$ is the output of activations in the k-th layer and $\mathbf{X}^{(0)} = \mathbf{X}$, $\mathbf{W}^{(k)}$ is the trainable weight matrix in the layer and σ is the activation function such as $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix added an self-loop to each node with graph Laplacian matrix $\widetilde{\mathbf{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$.

In (1), the graph convolution filter is

$$\mathbf{G}_o{=}\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2} = \mathbf{I}{-}\widetilde{\mathbf{L}}_\mathbf{s} \quad (2)$$

where $\widetilde{\mathbf{L}}_\mathbf{s}{=}\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{L}}\widetilde{\mathbf{D}}^{-1/2}$ is the symmetrically normalized Laplacian matrix of the graph represented by $\widetilde{\mathbf{A}}$. It was shown in [17] that the graph convolution of the GCN model is actually a special form of Laplacian smoothing, which is the key reason why GCN works. The new features of a node are computed as the weighted average of itself and its neighbors' by conducting Laplacian smoothing. The smoothing makes features of densely connected nodes in the graph similar, which makes them probably classified into the same class.

3. Proposed Approach

In this section, we first revisit GCN under the graph filtering framework, as studied in [18]. Then we introduce the designed graph convolutional filter that is compatible with video summarization. Finally, we present our detailed architecture for unsupervised or supervised learning of video summarization.

3.1. Graph Filtering

In graph signal processing [30], a linear convolutional graph filter can be represented by $\mathbf{G} = \mathbf{\Phi} p(\mathbf{\Lambda}) \mathbf{\Phi^{-1}} \in \mathbf{R^{N\times N}}$ where $p(\mathbf{\Lambda}) = \mathrm{diag}(p(\lambda_1), p(\lambda_2), ..., p(\lambda_N))$, $(\lambda_n)_{1\le n\le N}$ are the eigenvalues of Laplacian matrix in increasing order, and $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_N)$ is the matrix constructed by the associated orthogonal eigenvectors. The eigenvectors $(\boldsymbol{\phi}_n)_{1\le n\le N}$ are pairwise orthogonal and thus can form the Fourier basis. The associated eigenvalues $(\lambda_n)_{1\le n\le N}$ can be considered as frequencies and the function $p(\cdot)$ is the frequency response function of the filter $\mathbf{G}$. A graph signal $\mathbf{f}$ output a filtered signal $\bar{\mathbf{f}}$ after convolution with a graph filter $\mathbf{G}$, written as

$$\bar{\mathbf{f}} = \mathbf{G}\mathbf{f} = \mathbf{\Phi} p(\Lambda) \mathbf{\Phi}^{-1} \cdot \mathbf{\Phi}\mathbf{z} = \sum\nolimits_{n=1}^{N} p(\lambda_n) z_n \boldsymbol{\phi}_n \quad (3)$$

where $\mathbf{f}$ is decomposed into a linear combination of the Fourier basis written as

$$\mathbf{f} = \mathbf{\Phi}\mathbf{z} = \sum\nolimits_{n=1}^{N} z_n \boldsymbol{\phi}_n. \quad (4)$$

The magnitude of the coefficient $|z_n|$ represents the strength of the basis signal ϕ_n in $\mathbf{f}$, which is scaled by $p(\lambda_n)$ in the filtered signal $\bar{\mathbf{f}}$.

Each column of the feature matrix $\mathbf{X}$ can be considered as a graph signal. Graph structures and node features can be well integrated for learning by convoluting $\mathbf{X}$ with a graph filter $\mathbf{G}$. For standard GCN, the frequency response function of the graph filter defined in (2) is

$$p(\widetilde{\lambda}_n) = 1 - \widetilde{\lambda}_n \quad (5)$$

It is well known that high-frequency components are usually associated with noises, corresponding to large λ_n. To reduce noises and obtain smooth graph signals, the frequency response function $p(\cdot)$ should be low-pass. That is, the magnitude of $p(\cdot)$ should be larger for smaller λ_n and smaller for larger λ_n. However, the eigenvalues $(\widetilde{\lambda}_n)_{1\le n\le N}$ of the symmetrically normalized graph Laplacian $\widetilde{\mathbf{L}}_\mathbf{s}$ fall into interval [0,2] [4]. Obviously, the frequency response function in (5) is not low-pass for $\widetilde{\lambda}_n \in (1, 2]$ because its magnitude becomes larger for larger eigenvalues in $(1, 2]$, which will enhance high-frequency components and introduce noises.

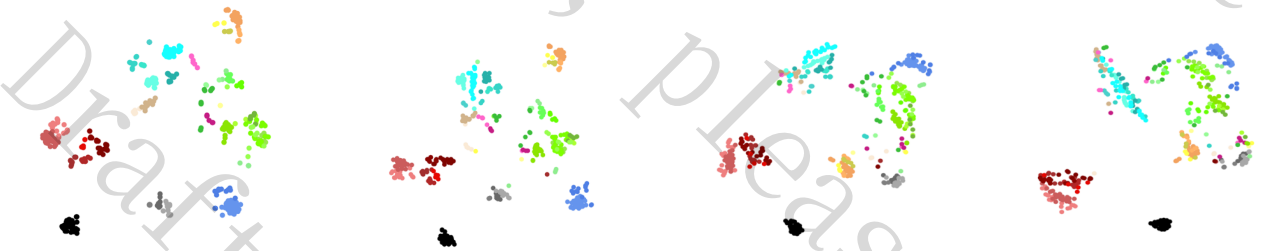
3.2. GCN with a Low-pass Graph Filter

The basic idea of our proposed GCN is to use the relation graph of video data to design a proper graph convolution filter, which can produce better data representations for the downstream video summarization task. We achieve this goal by applying a low-pass graph filter in GCN, which will produce a smooth signal. Here, we design the graph filter as

$$\mathbf{G}_s = \mathbf{I} - 0.5\widetilde{\mathbf{L}}_\mathbf{s} \quad (6)$$

whose frequency response function is

$$p(\widetilde{\lambda}_n) = 1 - 0.5\widetilde{\lambda}_n. \quad (7)$$



(a) Raw features (b) Output features of layer 1 (c) Output features of layer 2 (d) Output features of layer 3

Figure 2. T-SNE visualization of the (a) raw video features, and output features of (b) layer 1, (c) layer 2, (d) layer 3 in our encoder network. The selected video is Video 50 in TVSum.

The above frequency response function is strictly monotone decreasing for $\widetilde{\lambda}_n \in [0, 2]$ and behaves as a low-pass filter. This graph filter defined in (6) and its high-order forms have been used in [37] for graph clustering and achieved competitive performance. Applying such a low-pass graph filter on the feature matrix makes densely connected nodes have similar feature representations, and correspondingly nodes not connected closely will have more discriminative embeddings. The filtered features will exhibit a more compact cluster structure, which makes the downstream task of selecting keyframes from the whole video easier as verified in conventional cluster-based approaches.

For video summarization, we take each frame as a node and build the structural relation graph of video data by the cosine similarity of pairwise frame features, as

$$a_{i,j} = \frac{\mathbf{x}_i{}^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2} \tag{8}$$

where $a_{i,j}$ is (i, j)-th entry in the adjacency matrix $\mathbf{A}$, $\mathbf{x}_i$ is the feature embedding of i-th frame. Then the layer-wise propagation rule in GCN-LGF is

$$\mathbf{X}^{(k+1)} = \sigma(\mathbf{G}_s \mathbf{X}^{(k)} \mathbf{W}_s{}^{(k)}) \tag{9}$$

where $\mathbf{G}_s$ is defined in (6), $\mathbf{X}^{(k+1)}$ is the output features of the k-th layer and $\mathbf{X}^{(0)} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) \in \mathbf{R}^{\mathbf{N}\times\mathbf{c}}$ is the feature matrix of video frames, and $\mathbf{W}_s{}^{(k)}$ is the learnable weight matrix of the graph convolution in the k-th layer.

3.3. The SUM-GCN-LGF Architecture

The SUM-GCN-LGF architecture consists of an encoder network, a selector network and a decoder network, as shown in Figure 1. The encoder network contains three graph convolutional layers using the propagation rule defined in (9). The output filtered features of deeper graph convolutional layers may be oversmoothed and nodes from different clusters may become indistinguishable. Therefore, we propose a multiscale GCN-LGF as the encoder, as illustrated in Figure 1. The output features of the three GCN layers are concatenated as the input of the selector network. Figure 2(a)-(d) show the t-SNE visualization of the raw video features of Video 50 in TVSum [32], and output features of different layers in our encoder network. In Figure 2, frames in the same shot are represented by points in the same color, and shots in the same scene are represented by the same color hue but different luminance. The shot segments are obtained using Kernel Temporal Segmentation (KTS) [27] and the scenes are segmented artificially by human. It can be seen that as the network goes deeper, the output features are smoother and exhibit a more compact cluster structure. For videos having larger variation, deeper features are more useful, while for videos appear monotonous shallow features are more compatible. The network can obtain more information by the concatenated multiscale features for learning.

The concatenated multiscale features are fed into the selector network, consisting of two fully connected layers, as shown in Figure 1. The activation functions of the two layers are ReLU and sigmoid, respectively. The selector network outputs predicted scores represented by a vector $\mathbf{Y} = [y_1, y_2, ..., y_N]^T$, where $y_n \in [0, 1]$ indicates the probability of the n-th frame to be selected in a summary.

The raw frame features are weighted with the predicted scores and then forwarded to the decoder network. Our decoder network is composed of a two-layer GCN-LGF and outputs decoded features denoted by $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2, ..., \widehat{\mathbf{x}}_N]^T \in R^{N\times c}$. The reconstructed features can reconstruct a relation graph, presenting the recovered video structure. We define its adjacency matrix $\widehat{\mathbf{A}}$ by

$$\widehat{a}_{i,j} = \frac{\widehat{\mathbf{x}}_i^T \widehat{\mathbf{x}}_j}{\|\widehat{\mathbf{x}}_i\|_2 \|\widehat{\mathbf{x}}_j\|_2} \tag{10}$$

where $\widehat{a}_{i,j}$ is (i, j)-th entry in $\widehat{\mathbf{A}}$, representing the affinity between the i-th and j-th frame.

3.4. Loss Functions

Our unsupervised learning approach consists of three loss functions, i.e., a diversity loss, a reconstruction loss

and a sparsity loss as a regularization term.

Densely connected nodes usually come from the same scene. Therefore, we want to choose nodes as few as possible but at least one from these densely connected nodes to ensure the diversity and representativeness of generated summary. Considering that the selected frames in a diverse summary cannot have close affinity, we define the diversity loss as

$$\mathcal{L}_d = \frac{\sum_{i \neq j} \widehat{a}_{ij} y_i y_j}{\sum_{i \neq j} y_i y_j} \tag{11}$$

To ensure representativeness of the selected frames, the reconstruction loss is usually defined to reconstruct frame features, as

$$\mathcal{L}_c{}' = \frac{1}{N} \sum\nolimits_i \|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|_2^2 \tag{12}$$

However, we argue that raw and decoded video features can be represented in different embedding space. As long as the selected keyframes can well reconstruct video structural relation, they can present the complete storyline and should be considered as a desirable summary. Therefore, we design a new reconstruction loss as follows

$$\mathcal{L}_c = \frac{1}{N^2} \left\| \mathbf{A} - \widehat{\mathbf{A}} \right\|_1 \tag{13}$$

where $\|\cdot\|_1$ is an L_1 normalization.

We also add a regularization term to constrain the sparsity of the generated summary, as

$$\mathcal{L}_s = \left\| \frac{1}{N} \sum\nolimits_i y_i - \delta \right\|_1 \tag{14}$$

where δ is to control the percentage of frames selected for the summary.

Then the final loss function of the proposed SUM-GCN-LGF$_{unsup}$ approach for unsupervised learning becomes

$$\mathcal{L}_{unsup} = \mathcal{L}_d + \alpha \mathcal{L}_c + \beta \mathcal{L}_s \tag{15}$$

where α and β are used to control the trade-off among the three loss functions.

For supervised learning, we replace the sparsity constraint with a weighted binary cross entropy loss, formulated as

$$\mathcal{L}_{ce} = -\frac{1}{N} \sum\nolimits_i [(1-p) y_i^* \log(y_i) + p(1-y_i^*) \log(1-y_i)] \tag{16}$$

where y_i^* is the ground truth label of the i-th frame and p is the percentage of keyframes in a video. Thus, we can obtain the supervised loss of SUM-GDA-LGF as follows

$$\mathcal{L}_{sup} = \mathcal{L}_d + \alpha \mathcal{L}_c + \beta \mathcal{L}_{ce} \tag{17}$$

4. Experiments

We demonstrate the performance of the proposed SUM-GCN-LGF approach on two benchmark video summarization datasets, SumMe [9] and TVSum [32]. Firstly, we introduce some details of the datasets. Then, we present the implementation details as well as the experimental settings, which are followed by quantitative experimental results. Finally, the performance and analysis of several variants of our approach is presented.

4.1. Datasets

We evaluate the proposed SUM-GCN-LGF method on two publicly available benchmark datasets of video summarization, SumMe [9] and TVSum [32]. The SumMe dataset contains 25 user videos covering various topics such as holidays, sports, and history. The TVSum dataset contains 10 different categories collected from YouTube, including making sandwich, dog show, changing vehicle tire and so on. They are edited videos from the TRECVid Multimedia Event Detection (MED) task [31] and each category contains 5 videos. Each video in the two datasets is typically 1 to 5 minutes in length and is annotated by 15-20 users. SumMe provides annotations in the form of selected keyshots while TVSum is annotated in the form of frame-level importance scores. Both datasets provide a single ground-truth summary computed by averaging all users' scores, which is used for supervised learning. Following prior work [36, 19, 29, 38], we exploit 39 videos from the YouTube dataset [6] and 50 videos from the Open Video Project (OVP) dataset [1] to augment and transfer the training data.

4.2. Implementation Details and Experimental Settings

For fair comparison, we downsample videos by 2 fps and use the 1024-dimensional pool5 features of GoogleNet [33] pre-trained on ImageNet as the feature vector for each frame, following the majority of the state-of-the-art approaches. We train our model using the Adam optimizer with the initial learning rate 10^{-3} decayed by a factor 0.5 after 10 epochs. Early stopping is executed when the validate loss does not decrease for a period of time and the dropout rate is set to 0.5. We set $\alpha = 0.5$ and $\beta = 1$ in the loss function (15) and (17), which are optimized by cross-validation. The hyper-parameter δ in the sparsity loss is set as 0.15, following [38].

The keyshot-based metric commonly used in recent works is used for evaluation for a fair comparison. Similarity between a generated summary and a user-annotated summary is measured by the harmonic mean of precision and recall, i.e., F-score. The precision (P) and the recall

Table 1. Quantitative comparison (in terms of F-score %) between our supervised model and state-of-the-art supervised methods on SumMe [35] and TVSum [34].

	SumMe			TVSum		
	C	A	T	C	A	T
Dpp-LSTM [36]	38.6	42.9	41.8	54.7	59.6	58.7
SUM-GAN$_{sup}$ [19]	41.7	43.6	-	56.3	61.2	-
DR-DSN$_{sup}$ [38]	42.1	43.9	42.6	58.1	59.8	58.9
SUM-FCN [29]	47.5	51.1	44.1	56.8	59.2	58.2
VASNet [8]	49.7	51.1	-	61.4	62.4	-
ACGAN [11]	47.2	-	-	59.4	-	-
SumGraph [26]	51.4	52.9	**48.7**	**63.9**	**65.8**	**60.5**
SUM-GDA [16]	**52.8**	**54.4**	46.9	58.9	60.1	**59.0**
SUM-GCN-LGF	**52.4**	**54.0**	**47.6**	**62.8**	**63.1**	58.7

Table 2. Quantitative comparison (in terms of F-score %) between our unsupervised model and state-of-the-art unsupervised methods on SumMe [35] and TVSum [34].

	SumMe			TVSum		
	C	A	T	C	A	T
SUM-GAN$_{dpp}$ [19]	39.1	43.4	-	51.7	59.5	-
DR-DSN [38]	41.4	42.8	42.4	57.6	58.4	**57.8**
UnpairedVSN [28]	47.5	-	-	55.6	-	-
ACGAN [11]	46.0	47.0	44.5	58.5	58.9	**57.8**
SumGraph [26]	49.8	**52.1**	**47.0**	59.3	**61.2**	57.6
SUM-GDA$_{unsup}$ [16]	**50.0**	50.2	46.3	**59.6**	60.5	**58.8**
SUM-GCN-LGF$_{unsup}$	**52.7**	**53.0**	**46.5**	**59.4**	**60.6**	**57.8**

(R) is defined as follows,

$$P = \frac{M \cap U}{\|M\|} \tag{18}$$

$$R = \frac{M \cap U}{\|U\|} \tag{19}$$

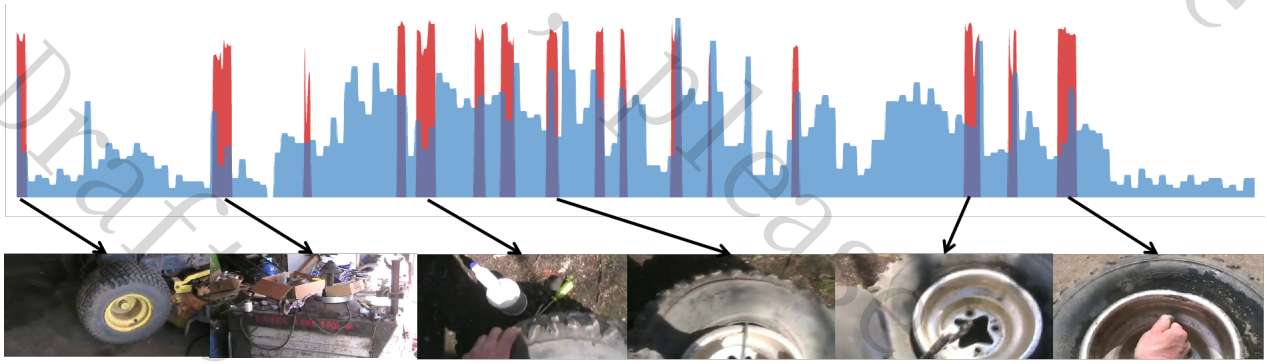
where M is a summary created by model and U represents a user summary.

To convert predicted keyframes to keyshots, we first apply KTS [27] to temporally segment videos into disjoint intervals, and a subset of intervals are selected by the 0/1 Knapsack algorithm under the constraint of not exceeding 15% in length of the video's duration.

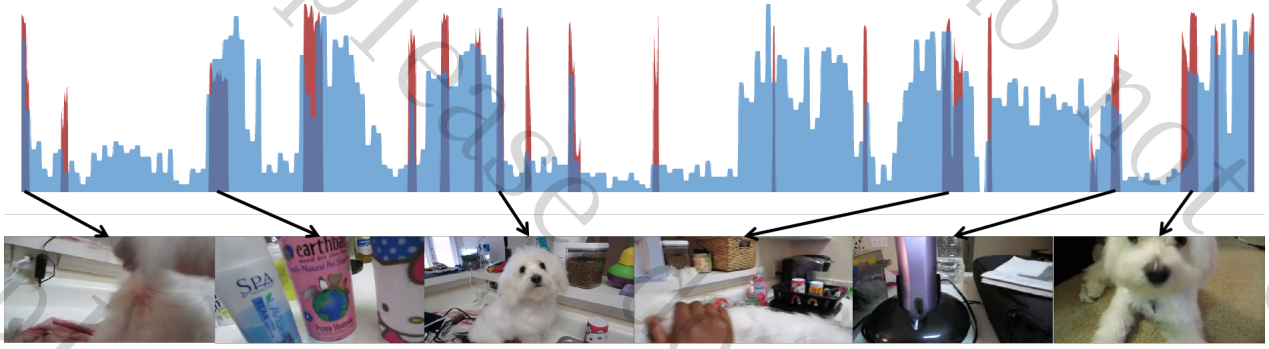
In the canonical (C) setting, we randomly select 80% for training and the remaining 20% for testing. We adopt five-fold cross validation and report the average performance. The other three datasets are added to supplement training data for augmented (A) learning. While in the transfer (T) setting, the model is trained on other available datasets and tested on all the videos in the given dataset. Following the protocol in recent works [36, 19, 29, 16, 26], we take the maximum of F-score over different human created summaries for SumMe, and computed the average of F-score for TVSum. All experiments are conducted 5 times and we report the average performance.

4.3. Quantitative Results

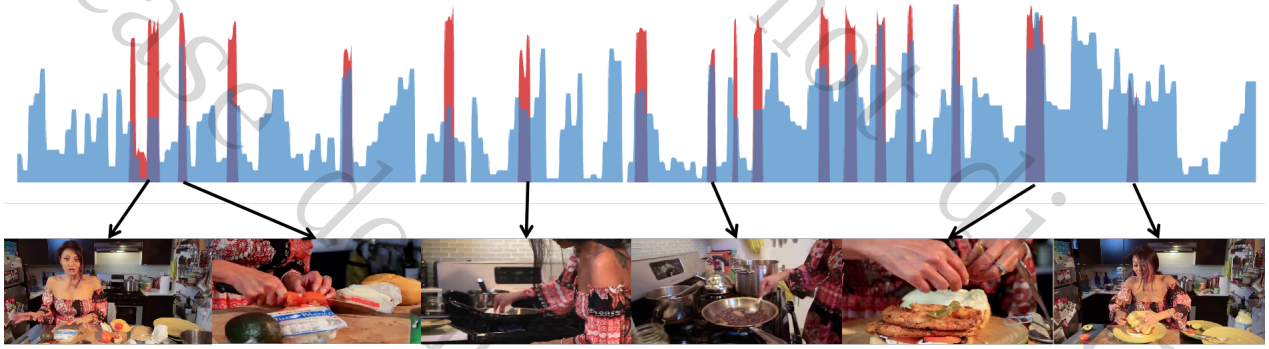
We compare the proposed model with other state-of-the-art supervised and unsupervised video summarization methods, as reported in Table 1 and Table 2. The best and second-best results are highlighted by bold fonts. As shown in the two tables, SUM-GCN-LGF either outperforms or is comparable to other state-of-the-art approaches. Especially, our unsupervised method outperforms other state-of-the art approaches by a large margin on SumMe in canonical and augmented settings. The SumGraph method and the SUM-GDA method also gain desirable performance. The former was also based on a GCN model, however, it refined the relation graph recursively for 5 iterations and brought high computation cost. The SUM-GDA method used global self-attention mechanism to model pairwise temporal relations among video frames. It mainly focused on generating diverse summaries, but ignored the representativeness of selected frames. Our method directly reasons on the relation graph, which can make full sense of temporal structure of video data. Moreover, the loss functions are designed to



(a) Video 1: Changing Tie



(b) Video 12: Puppy Dog Groomer



(c) Video 16: Sandwich Recipe

Figure 3. The architecture of SUM-GCN-LGF, consisting of a three-layer GCN-LGF as the encoder, a two-layer fully connected network taking in multiscale features as the selector, and a two-layer GCN-LGF as the decoder. Note that the objective of our decoder is to reconstruct video structures, not video features.

ensure both the diversity and the representativeness of the generated summaries. In Figure 3, we present the ground truth importance scores and the selected keyshots generated by our SUM-GCN-LGF model of different videos in TVSum, respectively represented by the blue bars and red bars. Six frames selected for the summary are presented below. As shown in Figure 3, the selected subshots capture almost all of the peak regions of the ground truth scores. The generated summary is visually diverse and close to the complete storyline conveyed by the original video, which demonstrates the effectiveness of our model.

4.4. Variants of our Approaches

In the proposed method, we improve GCN with a low-pass graph convolutional filter which is more compatible with the video summarization task. We conduct an additional experiment by replacing GCN-LGF with standard GCN. The results are presented in Table 3. Comparing the results of the second row with the last row, we can see that our model increases the F-score by 1% at least by using

Table 3. Comparison (in terms of F-score %) of different variations of our model in canonical setting for SumMe and TVSum datasets.

Method	SumMe	TVSum
SUM-GCN$_{unsup}$	51.0	58.6
SUM-GCN	51.2	61.8
SUM-GCN-LGF$_{ssf}$	51.4	62.1
SUM-GCN-LGF$_{cf}$	51.9	62.3
SUM-GCN-LGF	**52.4**	**62.8**

GCN-LGF. To verify the advantage of using multiscale features, we conducted an experiment by putting only the output features of layer 3 forward the selector netcwork. This variation using single scale features is named SUM-GCN-LGF$_{ssf}$. As shown in Table 3, multiscale features help to boost the performance by 1.0% on SumMe, and 0.7% on TVSum. In addition, our reconstruction loss is designed to preserve the structure of original videos in the generated summaries. By replacing the reconstruction loss with the conventional form, which is designed to reconstruct video features, we get a variant of our approach called SUM-GCN-LGF$_{cf}$. The structure preserving reconstruction loss outperforms the conventional reconstruction loss by 0.3% on SumMe, and 0.5% on TVSum. The superiority may be attributed to that the proposed loss function requires less constraint on the model, preventing overfitting in networks.

5. Conclusion

A novel video summarization model named SUM-GCN-LGF is proposed in this paper, which exploits a low-pass graph convolutional filter in GCN to obtain more compatible representations with the downstream video summarization task. To capture both low-level and high-level semantic information, multiscale features output by the three-layer GCN-LGF of the encoder network are utilized in the selector network. We also design a new reconstruction loss to preserve the structure of the original videos in the generated summaries. The SUM-GCN-LGF model directly reasons on the relation graph to learn the structural relationship of videos, which helps to produce high-quality summaries with the characteristic of diversity and representativeness. Quantitative experimental results demonstrate the effectiveness of our proposed approach.

References

[1] Open video project. [EB/OL]. https://open-video.org/. 5

[2] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. Unsupervised video summarization via attention-driven adversarial learning. In *International Conference on Multimedia Modeling*, pages 492–504. Springer, 2020. 1

[3] Sijia Cai, Wangmeng Zuo, Larry S Davis, and Lei Zhang. Weakly-supervised video summarization using variational encoder-decoder and web prior. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–200, 2018. 1

[4] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997. 3

[5] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2011. 1

[6] Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011. 5

[7] Pouriya Etezadifar and Hassan Farsi. Scalable video summarization via sparse dictionary learning and selection simultaneously. *Multimedia Tools and Applications*, 76(6):7947–7971, 2017. 2

[8] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Asian Conference on Computer Vision*, pages 39–54. Springer, 2018. 1, 6

[9] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014. 5

[10] Youssef Hadi, Fedwa Essannouni, and Rachid Oulad Haj Thami. Video summarization by k-medoid clustering. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1400–1401, 2006. 1, 2

[11] Xufeng He, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Unsupervised video summarization with attentive conditional generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2296–2304, 2019. 1, 2, 6

[12] Zhong Ji, Yuxiao Zhao, Yanwei Pang, Xi Li, and Jungong Han. Deep attentive video summarization with distribution consistency learning. *IEEE transactions on neural networks and learning systems*, 2020. 1

[13] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2698–2705, 2013. 1

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016. 1, 3

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

[16] Ping Li, Qinghao Ye, Luming Zhang, Li Yuan, Xianghua Xu, and Ling Shao. Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recognition*, 111:107677, 2020. 1, 6

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, pages 3538–3545, 2018. 3

[18] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label efficient semi-supervised learning via graph filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9582–9591, 2019. 3

[19] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017. 1, 2, 5, 6

[20] Karim M Mahmoud, Mohamed A Ismail, and Nagia M Ghanem. Vscan: an enhanced video summarization using density-based spatial clustering. In *International conference on image analysis and processing*, pages 733–742. Springer, 2013. 1, 2

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

[22] Shaohui Mei, Genliang Guan, Zhiyong Wang, Shuai Wan, Mingyi He, and David Dagan Feng. Video summarization via minimum sparse reconstruction. *Pattern Recognition*, 48(2):522–533, 2015. 2

[23] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006. 1, 2

[24] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Automatic video summarization by graph modeling. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 104–109. IEEE, 2003. 1, 2

[25] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K Roy-Chowdhury. Weakly supervised summarization of web videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3657–3666, 2017. 1

[26] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Sumgraph: Video summarization via recursive graph modeling. In *European Conference on Computer Vision*, 2020. 2, 6

[27] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014. 4, 6

[28] Mrigank Rochan and Yang Wang. Video summarization by learning from unpaired data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7902–7911, 2019. 1, 2, 6

[29] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 347–363, 2018. 1, 2, 5, 6

[30] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013. 3

[31] Alan F Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330, 2006. 5

[32] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015. 4, 5

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[34] Jiaxin Wu, Sheng-hua Zhong, Jianmin Jiang, and Yunyun Yang. A novel clustering method for static video summarization. *Multimedia Tools and Applications*, 76(7):9625–9641, 2017. 1, 2

[35] Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9143–9150, 2019. 1, 2

[36] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016. 1, 2, 5, 6

[37] Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. Attributed graph clustering via adaptive graph convolution. pages 4327–4333, 08 2019. 4

[38] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 7582–7589, 2018. 1, 2, 5, 6

[39] Yueting Zhuang, Yong Rui, Thomas S Huang, and Sharad Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, volume 1, pages 866–870. IEEE, 1998. 1