SumGraph: Video Summarization via Recursive Graph Modeling

Jungin Park[1⋆], Jiyoung Lee[1⋆], Ig-Jae Kim[2], and Kwanghoon Sohn[1⋆⋆]

[1] Yonsei University, Seoul, Korea
[2] Korea Institute of Science and Technology(KIST), Seoul, Korea
{newrun, easy00, khsohn}@yonsei.ac.kr
drjay@kist.re.kr

Abstract. The goal of video summarization is to select keyframes that are visually diverse and can represent a whole story of an input video. State-of-the-art approaches for video summarization have mostly regarded the task as a frame-wise keyframe selection problem by aggregating all frames with equal weight. However, to find informative parts of the video, it is necessary to consider how all the frames of the video are related to each other. To this end, we cast video summarization as a graph modeling problem. We propose recursive graph modeling networks for video summarization, termed SumGraph, to represent a relation graph, where frames are regarded as nodes and nodes are connected by semantic relationships among frames. Our networks accomplish this through a recursive approach to refine an initially estimated graph to correctly classify each node as a keyframe by reasoning the graph representation via graph convolutional networks. To leverage SumGraph in a more practical environment, we also present a way to adapt our graph modeling in an unsupervised fashion. With SumGraph, we achieved state-of-the-art performance on several benchmarks for video summarization in both supervised and unsupervised manners.

Keywords: video summarization, graph convolutional networks, recursive graph refinement

1 Introduction

Over the years, the growth of online video platforms has made it difficult for users to access the video data they want. Moreover, the length of the uploaded videos is getting more extended every day, and it can be impractical for people to watch these videos in full to obtain useful information. In response to these issues, computer vision techniques have attracted intense attention in recent years for efficient browsing of the enormous video data. In particular, the research topic, which automatically selects a simple yet informative summary that succinctly

⋆ Both authors contributed equally to this work
⋆⋆ Corresponding author

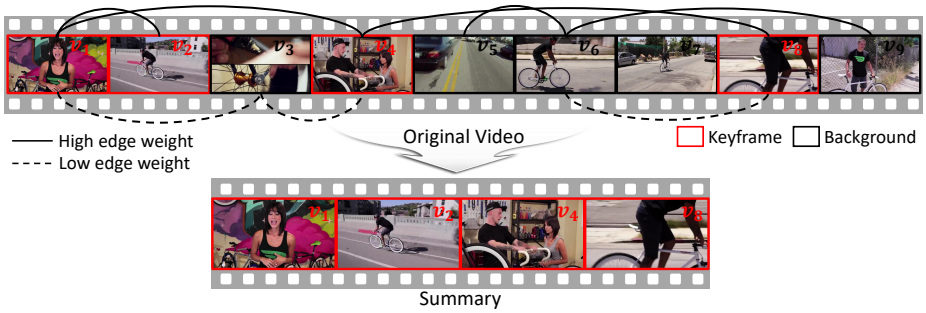


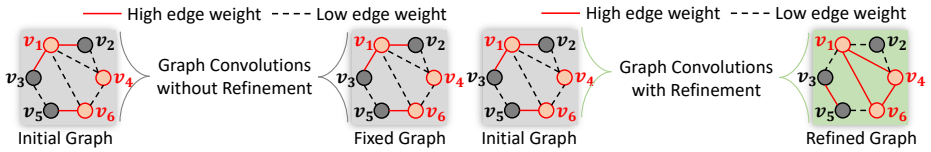
Fig. 1. Illustration of SumGraph. We regard video summarization as a graph modeling problem. We obtain a richer video summary by constructing the graph represents relationships between frames in a video and recursively refining the graph.

depicts the contents of the original video, has become a prominent research topic as a promising tool to cope with the overwhelming amount of video data [40,21].

Inspired by the great successes of deep learning in recent years, current approaches [41,21,10,37] have commonly treated video summarization as a sequence labeling or scoring problem to solve it with variant of recurrent neural networks (RNNs). Although RNNs efficiently captures long-range dependencies among frames, the operations of RNNs are applied repeatedly, propagating signals progressively through the frames. It causes optimization difficulties that need to be carefully addressed [9], as well as multihop dependency modeling, which makes it difficult to deliver messages back and forth [34]. To tackle these limitations, Rochan *et al.* [30] proposed fully convolutional sequence networks (SUM-FCN), in which video summarization is considered as a sequence labeling problem. However, it inevitably neglects semantic relevance between keyframes with varying time distances.

Graphical models have been used to specifically model semantic interactions [17,2,16,39,11,25]. These approaches have been recently revisited with graph convolutional networks (GCNs) [16], which generalize convolutions from grid-like data to non-grid structures. GCNs have therefore been the subject of increasing interest in various computer vision applications, such as object detection [38], video classification [35], video object tracking [5], and action localization [39]. These works model knowledge graphs based on the relationships between different entities, such as images, objects, and proposals. Although GCNs have shown promising results, they generally use a fixed graph directly obtained from the affinity of feature representations [12], where nodes are strongly connected when they have similar entities. These approaches are therefore difficult to be directly employed for video summarization. It requires to extract the comprehensive semantic relationships between frames which identify the connections of the whole story in the video to infer the summary.

In this paper, we cast video summarization as a relation graph modeling problem. We present a novel framework, referred as SumGraph, to incorporate the advantages of graph modeling into a deep learning framework, which is suited



(a) Prior works (b) Our approach

Fig. 2. Comparison of our approach with prior works using GCNs. The red circle and black circle represent node features of keyframes and background, respectively: (a) methods using GCNs without graph refinement and (b) SumGraph, which refines an initial graph by recursively estimating the graph. With the graph refinement, SumGraph leverages semantic relationships.

to the modeling of frame-to-frame interactions. As illustrated in Fig. 1, we regard a video frame as a node of a relation graph. Nodes are connected by edges representing semantic affinities between nodes to represent the relationships over frames in a video. If a node is included in a summary, it is connected to other keyframe nodes with high affinity weights, so that the semantic connections between frames can be modeled through graph convolutions. We additionally formulate SumGraph to recursively estimate the relation graph, which is used for iterative reasoning of graph representations. As shown in Fig. 2, our approach leverages semantic relationships between nodes by recursively estimating and refining the relation graph. In contrast to prior works using a fixed graph without refinement, SumGraph enhances the graph and refines the feature representations according to the updated graph. The proposed method is extensively examined through an ablation study, and comparison with previous methods in both a supervised and an unsupervised manner on two benchmarks including SumMe [8] and TVSum [31].

Our contributions are as follow:

- To the best of our knowledge, this work is the first attempt to exploit deep graph modeling to perform relationship reasoning between frames for video summarization.
- Our model recursively refines the relation graph to obtain the optimal relation graph of an input video, and leverages the estimated graph to classify whether a node is a keyframe.
- The experimental results show that the presented approach achieves state-of-the-art performance in both supervised and unsupervised manner.

2 Related Work

2.1 Video Summarization

Given an input video, video summarization shortens an original video into a short watchable synopsis, resulting in outputs such as video synopses [28], time-

lapses [13,26], montages [14,32], or storyboards [6,7]. Early video summarization relied primarily on handcrafted criteria [14,18,19,22,20,27], including importance, relevance, representativeness and diversity to produce a summary video.

Deep networks have recently been applied with considerable success, CNN based approaches have made significant progress [30,42,41]. Recurrent models are widely used for video summarization capturing variable range dependencies between frames [41,42]. Although recurrent network based approaches have been applied to video data, recurrent operations are sequential, limiting the processing all the frames simultaneously. To overcome this limitation, Rochan *et al.* [30] considered video summarization as a binary label prediction problem, by establishing a connection between semantic segmentation and video summarization. However, consideration of modeling the relationship amongst frames, which provides significant cues as to how best to summarize the video, is not addressed in these approaches. Fajtl *et al.* [4] proposed a self-attention based video summarization method that modeled pairwise relations between frames. While they showed significant performance improvements, they considered visual similarity only without considering semantic similarity (*i.e.*, keyframe and background).

Existing methods have been extended into an unsupervised training scheme [37,21,29]. Mahasseni *et al.* [21] extended an LSTM based framework with a discriminator network without human annotated summary videos. Their frame selector uses a variational auto encoder LSTM to decode the output for reconstruction through selected frames, and the discriminator is an other LSTM network that learns to distinguish between the input video and its reconstruction. Rochan *et al.* [29] trained a network from unpaired data using adversarial learning. While He *et al.* [10] produced weighted frame features to predict importance scores with attentive conditional GANs, Yuan *et al.* [37] proposed a cycle consistent learning objective to relieve the difficulty of unsupervised learning. Although these approaches resolved the problem of insufficient data, the locality of recurrent and convolutional operations, which does not directly compute relevance between any two frames, is still problematic.

2.2 Graphical Models

Our notion of modeling a graph from video is partly related to recent research in graphical models. One popular direction is using conditional random fields (CRF) [17], especially for semantic segmentation [17,2], where the CRF model is applied to all pairs of pixels in an image to infer mean-field with high confidence.

An attempt at modeling pairwise spatiotemporal relations has been made in the non-local neural networks [34]. However, the model is not explicitly defined on graphs. Moreover, the non-local operator is applied to every pixel in the feature space, from lower to higher layers incurring a high computational cost. Graph convolutional networks (GCNs) [16] have been used in several areas of computer vision such as skeleton-based action recognition [36], video classification [11,35], visual object tracking [5], and temporal action localization [39]. They have demonstrated the effectiveness of GCNs by exploiting the relations between input data, and have shown satisfactory performance on each task. However, an

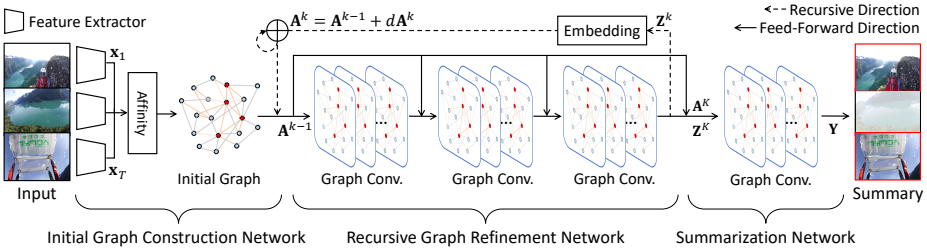


Fig. 3. Network configuration of SumGraph, consisting of a initial graph construction network, recursive graph refinement network, and summarization network in a recurrent structure.

initial graph obtained directly from the affinity of input feature representations is suboptimal graph [12]. Rather than using the initially obtained graph as a fixed form in GCNs, we recursively refine the graph model by learning the relationship between keyframes to obtain the optimal relation graph.

3 Preliminaries

Here we present a brief review of GCNs [16]. Given a graph $\mathcal{G}$ represented by a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of unordered vertices and $\mathcal{E}$ is the set of edges representing the connectivity between vertices $v \in \mathcal{V}$. GCNs aim to extract richer features at a vertex by aggregating the features of vertices from its neighborhood. The vertices v_i and v_j are connected to each other with an edge $e_{ij} \in \mathcal{E}$. GCNs represent vertices by associating each vertex v with a feature representation h_v. The adjacency matrix $\mathbf{A}$ is derived as a $T \times T$ matrix with $A_{ij} = 1$ if $e_{ij} \in \mathcal{E}$, and $A_{ij} = 0$ if $e_{ij} \notin \mathcal{E}$.

In standard GCNs, the output of the graph convolution operation is as follows:

$$\mathbf{Z} = \sigma(\mathbf{D}^{-1/2}\hat{\mathbf{A}}\mathbf{D}^{-1/2}\mathbf{X}\mathbf{W}), \tag{1}$$

where $\sigma(\cdot)$ denotes an activation function such as the ReLU, $\mathbf{X}$ and $\mathbf{Z}$ are input and output features, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{D}$ is a diagonal matrix in which a diagonal entry is the sum of the row elements of $\hat{\mathbf{A}}$, and $\mathbf{W}$ is a trainable weight matrix in the graph convolution layer, respectively. Given the graph representation, we can perform reasoning on the graph by applying the GCNs, rather than applying CNNs or RNNs, which have limited capability to represent relationships between features.

4 Recursive Graph Modeling Networks

4.1 Motivation and Overview

In this section, we describe the formulation of recursive graph modeling networks, termed SumGraph. Inspired by [22], our networks learn to model the input video

as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to select a set of keyframes, $\mathcal{S}$, where each frame is treated as a node, $v \in \mathcal{V}$, and each edge, $e \in \mathcal{E}$, is used to represent the relation between frames.

In conventional GCNs, the input graph is constructed and fixed through input data with explicit graphical modeling. However, we seek to iteratively refine an initial graph in order that the nodes are connected by the story of the video, not the similarity between entities. To realize this, we formulate recursive graph modeling networks which gradually complete the optimal graph, by repeatedly estimating the adjacency matrix $\mathbf{A}$. Our networks are split into three parts, including an *initial graph construction network* to build the initial graph $\mathcal{G}$, a *recursive graph refinement network* to infer the optimal graph incorporating the semantic relationships between frames, and a *summarization network* to classify the node features into keyframes for the summary video, as shown in Fig. 3.

4.2 Network Architecture

Initial graph construction network. To obtain an initial graph, we first compute the affinity between every pair of frame features. The edge weights of the initial graph are set using the affinity scores. The affinity between two frame features is computed as their cosine similarity:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{||\mathbf{x}_i||_2 \cdot ||\mathbf{x}_j||_2}, \tag{2}$$

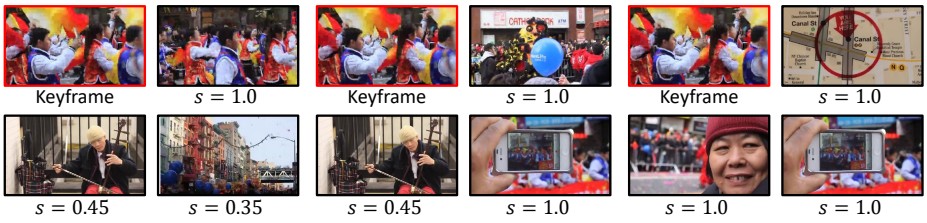
where $\mathbf{x} \in \mathbf{X}$ is a node feature of $\mathcal{V}$. We assign $f(\mathbf{x_i}, \mathbf{x}_j)$ to the (i, j)-th entry in the adjacency matrix of the initial graph $\mathbf{A}^0$. The initial graph is passed to a subsequent graph convolution layer with input feature $\mathbf{X}$,

$$\mathbf{X}^0 = \sigma(\tilde{\mathbf{A}}^0 \mathbf{X} \mathbf{W}_C), \tag{3}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$ and $\mathbf{W}_C$ is the learnable weight matrix of the graph convolution layer. The output of the graph convolution layer is an aggregated feature from its neighborhoods for each node. We denote the node number as T, thus the adjacency matrix dimension has dimensionality $T \times T$.

Recursive graph refinement network. Constructing edges by linking all frames with each other will aggregate the redundant and noisy information for video summarization. Therefore, the connection between semantically unrelated frames should be disconnected. Formally, given the graph and features, an intermediate feature denoted as $\mathbf{Z}^k$ is extracted by a feed-forward graph convolution process such that, $\mathbf{Z}^k = \mathcal{F}(\mathbf{Z}^{k-1}, \mathbf{A}^{k-1}|\mathbf{W}_R)$ with the network parameters $\mathbf{W}_R$. The recursive graph refinement network repeatedly estimates the residual between the previous and current adjacency matrix as

$$\begin{aligned} \mathbf{A}^k - \mathbf{A}^{k-1} &= d\mathbf{A}^k \\ &= \frac{(\mathbf{W}_\theta \mathbf{Z}^k)^T (\mathbf{W}_\phi \mathbf{Z}^k)}{||\mathbf{W}_\theta \mathbf{Z}^k||_2 \cdot ||\mathbf{W}_\phi \mathbf{Z}^k||_2}, \end{aligned} \tag{4}$$



(a) Iteration 1 (b) Iteration 3 (c) Iteration 5

Fig. 4. Convergence of SumGraph: Frames which have top3 affinity value with a selected keyframe at (a) iteration 1; (b) iteration 3; and (c) iteration 5, where s denotes the normalized and averaged user-annotated importance scores which range from 0 to 1. As the graph refinement repeats in SumGraph, the keyframes are progressively connected with high affinity value.

where $\mathbf{W}_\theta$ and $\mathbf{W}_\phi$ are two different weight matrices to be learned in the network, that enable to compute the affinity in a linear embedding space. The final adjacency matrix is then estimated in a recurrent manner as follows:

$$\mathbf{A}^K = \mathbf{A}^0 + \sum_{k=1}^{K} d\mathbf{A}^k, \tag{5}$$

where K denotes the maximum iteration and $\mathbf{A}^0$ is an initial adjacency matrix from the initial graph construction network.

In contrast to [35], in which the affinity scores of input features were considered as a fixed adjacency matrix, we obtain the optimal adjacency matrix using an iterative refinement procedure with output feature representations. Repeatedly inferring the residuals of the adjacency matrix facilitates fast convergence for video summarization. Moreover, frames initially connected by the visual similarity are progressively linked into the subset of frames based on the semantic connection. As shown in Fig. 4, the edge weights between semantically connected frames (*i.e.*, keyframes) become progressively higher through iterative estimation.

Summarization network. As illustrated in Fig. 3, the updated node features of the refined optimal graph after the recursive graph refinement network are fed into a summarization network for graph reasoning. Our final goal is to classify each node in the optimal graph that is linked by the semantic relationships between frames. We append a graph convolutional layer, followed by a sigmoid operation to obtain a summary score $\mathbf{Y}$ indicating whether each node is included in summary such that:

$$\mathbf{Y} = \mathcal{F}(\mathbf{Z}^K, \mathbf{A}^K | \mathbf{W}_S) \tag{6}$$

with matrix parameters $\mathbf{W}_S$. Similar to the binary node classification tasks, if the summary score of each node $y_i \in \mathbf{Y}$ is higher than 0.5, the i-th node is selected for keyframes such that $v_i \in \mathcal{S}$.

4.3 Loss Functions

To deal with the imbalance between the number of keyframes and the number of background frames, we design two loss functions., node classification loss and sparsity loss. We define node classification loss as a weighted binary cross entropy loss for supervised learning [30]:

$$\mathcal{L}_c = -\frac{1}{T}\sum_{t=1}^{T} w_t[y_t^* \log(y_t) + (1-y_t^*)\log(1-y_t)], \tag{7}$$

where y_t^* is the groundtruth label of the t-th frame. Each node is weighted by $w_t = median\_freq/freq(c)$ for the t-th frame. In our work, $freq(c)$ is $\frac{|\mathcal{S}_v|}{T}$ for keyframes and $1-\frac{|\mathcal{S}_v|}{T}$ for background, where $|\mathcal{S}_v|$ is the number of keyframes in video v. Since the number of classes is 2 (keyframe or not), $median\_freq$ is set to 0.5.

In practice, a long video can be summarized into a sparse subset of keyframes. Based on this intuition, SumGraph learns parameters with which to construct sparse connections between graph nodes for video summarization. To enforce this constraint, the sparsity loss is given by an L_1 normalization that measures the sparsity of node connections on the final adjacency graph as follows [23]:

$$\mathcal{L}_s = \sum_{i=1}^{T}\sum_{j=1}^{T} ||a_{ij}||_1, \tag{8}$$

where $a_{ij} \in \mathbf{A}^K$ is the affinity between two nodes, v_i and v_j.

For diverse keyframe selection [41,30], we use additional loss functions such as reconstruction and diversity loss. We apply additional graph convolutions to selected keyframes $\mathbf{Y}_\mathcal{S}$ to reconstruct the original features $\mathbf{X}_\mathcal{S}$. We use two graph convolution layers so that the dimensionality of the reconstructed features is the same as that of the original features. The reconstruction loss $\mathcal{L}_r$ is defined as the mean squared error between the reconstructed features and the original features, such that:

$$\mathcal{L}_r = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} ||\mathbf{x}_i - \hat{\mathbf{y}}_i||_2^2, \tag{9}$$

where $\hat{\mathbf{y}}$ denotes the reconstructed features.

From [30,29], we employ a repelling regularizer [43] as the diversity loss, $\mathcal{L}_d$, to enforce the diversity of selected keyframes:

$$\mathcal{L}_d = \frac{1}{|\mathcal{S}|(|\mathcal{S}|-1)}\sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}, j\neq i} f(\hat{\mathbf{z}}_i, \hat{\mathbf{z}}_j), \tag{10}$$

where $f(\cdot)$ is the affinity function in (2), $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_j$ denote the reconstructed feature vectors of the i-th and j-th node.

The final loss function for supervised learning is then,

$$\mathcal{L}_{sup} = \mathcal{L}_c + \lambda\cdot\mathcal{L}_s + \alpha\cdot\mathcal{L}_d + \beta\cdot\mathcal{L}_r, \tag{11}$$

Method	SumMe			TVSum		
	Standard	Augment	Transfer	Standard	Augment	Transfer
Zhang *et al.* [40]	40.9	41.3	38.5	-	-	-
Zhang *et al.*[41]	38.6	42.9	41.8	54.7	59.6	58.7
Mahasseni *et al.* [21] (SUM-GAN$_{sup}$)	41.7	43.6	-	56.3	61.2	-
Rochan *et al.* [30](SUM-FCN)	47.5	51.1	44.1	56.8	59.2	58.2
Rochan *et al.* [30](SUM-DeepLab)	48.8	50.2	45.0	58.4	59.1	57.4
Zhou *et al.* [44]	42.1	43.9	42.6	58.1	59.8	58.9
Zhang *et al.* [42]	-	44.9	-	-	63.9	-
Fajtl *et al.* [4]	49.7	51.1	-	61.4	62.4	-
Rochan *et al.* [29]	-	48.0	41.6	-	56.1	55.7
He *et al.* [10]	47.2	-	-	59.4	-	-
Ours	**51.4**	**52.9**	**48.7**	**63.9**	**65.8**	**60.5**

Table 1. Comparison of our algorithm with other recent supervised techniques on the SumMe [8] and TVSum [31] datasets, with various data configurations including standard data, augmented data, and transfer data settings.

Method	SumMe			TVSum		
	Standard	Augment	Transfer	Standard	Augment	Transfer
Mahasseni *et al.*[21]	39.1	43.4	-	51.7	59.5	-
Yuan *et al.*[37]	41.9	-	-	57.6	-	-
Rochan *et al.*[30](SUM-FCN$_{unsup}$)	41.5	-	39.5	52.7	-	-
Rochan *et al.*[29]	47.5	-	41.6	55.6	-	55.7
He *et al.*[10]	46.0	47.0	44.5	58.5	58.9	**57.8**
Ours	**49.8**	**52.1**	**47.0**	**59.3**	**61.2**	57.6

Table 2. Comparison of our algorithm with other recent unsupervised techniques on the SumMe [8] and TVSum [31] datasets with standard data, augmented data, and transfer data configurations.

where λ, α and β control the trade-off between the four loss functions. The graph modeling scheme in SumGraph has also been extended for unsupervised video summarization with simple modifications to the loss functions. Since the groundtruth summary cannot be used for supervision in an unsupervised manner, the final loss function for unsupervised learning is represented as:

$$\mathcal{L}_{unsup} = \mathcal{L}_s + \alpha \cdot \mathcal{L}_d + \beta \cdot \mathcal{L}_r, \tag{12}$$

where α and β are balancing parameters to control the trade-off between the three terms.

5 Experimental Results

5.1 Implementation Details

To train SumGraph, we uniformly sample frames for every video at 2 fps, as described in [41]. For feature extraction, we use the ImageNet-pretrained

Method	SumMe			TVSum		
	Precision	Recall	F-score	Precision	Recall	F-score
Rochan *et al.*[30]†* (SUM-FCN$_{unsup}$)	43.9	46.2	44.8	59.1	49.1	53.6
Rochan *et al.*[29]†* (UnpairedVSN)	46.3	49.4	46.5	61.1	50.9	55.6
Rochan *et al.*[29]‡* (UnpairedVSN$_{psup}$)	46.7	49.9	48.0	61.7	51.4	56.1
Ours†	48.2	51.1	49.6	59.7	58.9	59.3
Ours‡	**50.6**	**52.3**	**51.4**	**64.3**	**63.5**	**63.9**

Table 3. Summarization performance (%) on the SumMe [8] and TVSum [31] in terms of three standard metrics: Precision, Recall, and F-score. † denotes an unsupervised method and ‡ denotes supervised method. $*$ is taken from [29].

Method	Kendall's τ	Spearman's ρ
Zhang *et al.* [41]	0.042	0.055
Zhou *et al.* [44]	0.020	0.026
Human	0.177	0.204
Ours	0.094	0.138

Table 4. Kendall's τ [15] and Spearman's ρ [45] correlation coefficients computed on the TVSum benchmark [31].

GoogleNet [33], where the 1024-dimensional activations are extracted from the 'pool5' layer. Note that our feature extraction follows prior work [30,42], in order to ensure fair comparisons.

Since different datasets provide groundtruth annotations in various formats, we follow [30,6,41] to generate single keyframe based annotations. If a frame is selected for summary video, label it 1; otherwise, label it 0. During testing, we follow [41,30] to convert predicted keyframes to keyshots, to allow fair comparison with other methods.

We train our model for 50 epochs using Adam optimizer with a batch size of 5. The learning rate is set to 10^{-3} and decayed by a factor of 0.1 for every 20 epochs. We set $\lambda = 0.001$, $\alpha = 10$ and $\beta = 1$ in (11) for supervised learning, and $\alpha = 100$ and $\beta = 10$ in (12) for unsupervised learning. The maximum iteration number K is fixed to 5 throughout the ablation study. All experiments are conducted five times on five random splits of the data, and we report the average performance. Details of the implementation and training of our system are provided in the supplemental material. Our code will be made publicly available.

5.2 Experimental Settings

Datasets. We evaluate our approach on two standard video summarization datasets: SumMe [8] and TVSum [31]. SumMe dataset consists of 25 videos capturing multiple events such as cooking and sports, and the lengths of the videos vary from 1.5 to 6.5 minutes. The TVSum dataset contains 10 categories from the MED task, and samples 5 videos per category from YouTube. The

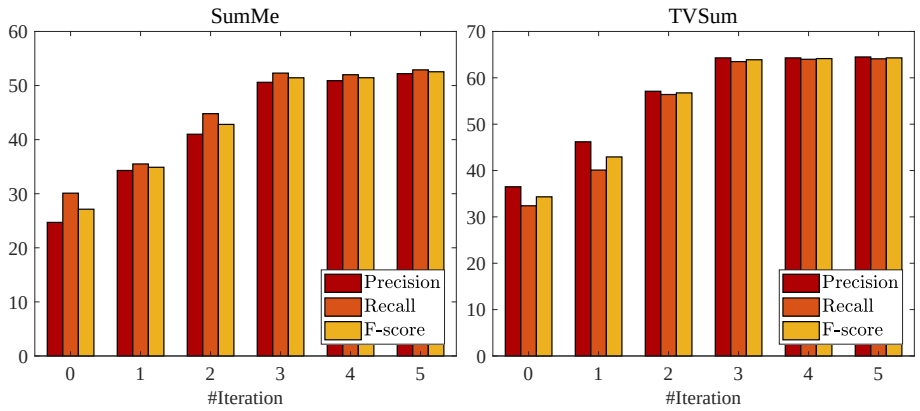


Fig. 5. Convergence analysis with respect to the number of iterations on (a) SumMe benchmark [8] and (b) TVSum benchmark [31].

contents of the videos are diverse, similar to SumMe, and the video lengths vary from one to five minutes. Those datasets provide frame-level importance scores annotated by several users. We use additional datasets, the YouTube [3] and the OVP [1] datasets, to augment the training data.

Data configuration. We conduct experiments using different three data configurations: standard supervision, augmented data setting, and transfer data setting. In standard data setting, the training and testing videos are from the same dataset. We randomly select 20% of videos for testing and the rest of the videos are used for training and validation. For the augmented data setting, we used the other three datasets to augment the training data. By augmenting the training data, recent works [42,30] showed improved performance, and our experimental results derived similar conclusion. In the transfer data setting, which is a more challenging setting introduced in [40,41], we train our models on other available datasets, and the given dataset is used only for evaluating performance. Note that all videos in a given dataset are used for evaluation only.

Evaluation metrics. We evaluate our method using the keyshot-based metrics commonly used in recent works [30,29]. Let Y and Y^* be the predicted keyshot summary and the groundtruth summary created by multiple users, respectively. We define the precision and the recall as follows:

$$\begin{aligned} Precision &= \frac{\text{overlap between } Y \text{ and } Y^*}{\text{total duration of } Y}, \\ Recall &= \frac{\text{overlap between } Y \text{ and } Y^*}{\text{total duration of } Y^*} \end{aligned} \tag{13}$$

We compute the F-score to measure the quality of the summary with the precision and the recall:

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Presion + Recall}. \tag{14}$$

For datasets with multiple groundtruth summaries, we follow standard approaches [41,8,7] to calculate the metrics for the videos.

We also evaluate our method using the rank based metrics, Kendall's τ [15] and Spearman's ρ [45] correlation coefficients, following [24]. To compute the correlation coefficients, we first rank the video frames according to their probability of being a keyframe and the annotated importance scores. And we compare the generated ranking with each annotated ranking. The correlation scores are then computed by averaging over the individual results.

5.3 Results

In Table 1, we show our results in comparison to supervised video summarization methods [40,41,21,42,30,4,29,10] in terms of F-score on the SumMe and TVSum datasets. We observed a significant boost in performance over the state-of-the-art methods in various data configuration settings. Our model achieves state-of-the-art performance of 52.9% on the SumMe and 65.8% on the TVSum datasets in the data augmented setting.

We compare our results trained in an unsupervised manner with recent unsupervised approaches for video summarization [31,21,30,37,29,10] in Table 2. While the results showed inferior performance to the supervised approaches, the use of our relation graph improved the performance without unpaired data. The results show that our sparsity and diversity losses are sufficient to learn the graphical model for video summarization in unsupervised manner.

Table 3 shows the performance according to precision, recall, and F-score. Our method outperforms supervised approaches for video summarization on all evaluation metrics. Especially, results show that the improvement in recall is greater than the improvement in precision, as SumGraph provides a more accurate summary. Surprisingly, the performance of our unsupervised approach even outperforms the approach using partial supervision (UnpairedVSN$_{psup}$). The comparison studies demonstrate that graphical modeling using SumGraph is an effective approach to the development summary video in comparison with CNNs and RNNs based approaches.

The results for the correlation coefficients are shown in Table 4. We compare our results with those results of two methods, deepLSTM [41] and DR-DSN [44], reported in [24]. Although we use the probabilities of the keyframes, not the importance scores, our model outperforms the existing models by 0.052 for Kendall's τ and 0.083 for Spearman's ρ.

5.4 Ablation Study

Number of iterations. All ablation studies are investigated with standard supervision configuration with the TVSum benchmark [31]. To validate the ef-

Classification	Sparsity	Diversity + Reconstruction	F-Score
✓			62.8
✓	✓		63.6
✓		✓	63.5
✓	✓	✓	63.9

Table 5. Ablation study for the various combination of loss functions in SumGraph on the TVSum benchmark [31].

fectiveness of the recursively estimate relation graph used in SumGraph, we examined the F-score corresponding to the number of iterations. With increasing numbers of iterations, the accuracies of all evaluation metrics gradually improved. SumGraph converges in three to five iterations as shown in Fig. 5. The additional results of ablation study for the number of iterations are provided in the supplemental material.

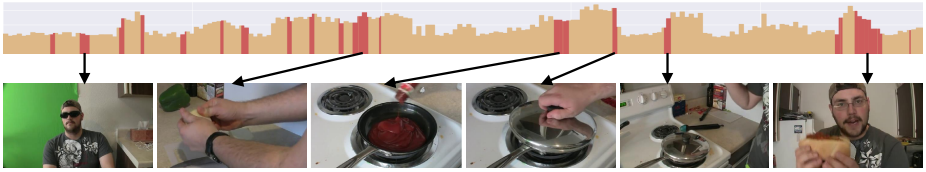
Loss functions. We conducted an additional experiment to investigate the effectiveness of the loss functions. When we trained our model in a supervised manner, four loss terms were employed, classification, sparsity, reconstruction, and diversity losses. The reconstruction loss forces the reconstructed features to be similar to the original features, and the diversity loss enforces the diversity of reconstructed features. We analyzed the effectiveness of the reconstruction and diversity losses together. To verify the effectiveness of each loss function, we applied the diversity and the reconstruction losses, and compared their results in a supervised setting. As shown in Table 5, we observed that both the sparsity loss and the sum of the diversity and the reconstruction loss improve the performance by 0.8% and 0.7%, respectively, on the TVSum benchmark. The full usage of loss functions contributes a performance improvement of 1.1%.

5.5 Qualitative Analysis

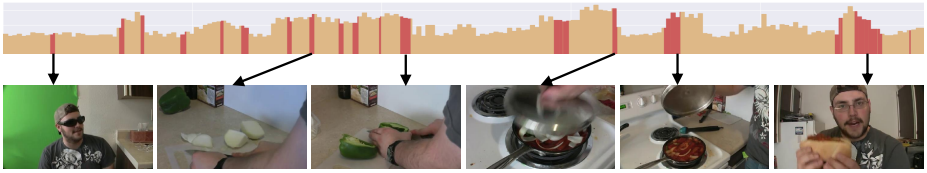
In Fig. 6, we present the groundtruth importance scores and selected frames produced by [30] and SumGraph. For the visualization of the summaries, we sampled six frames which had been selected for summary. The red bars on brown backgrounds are the frames selected as summaries. The summary generated by SumGraph is visually more diverse, and captures almost all of the peak regions of the groundtruth scores. This observation indicates that SumGraph is able to estimate semantic relationships and connects informative frames for generating optimal and meaningful summaries.

6 Conclusion

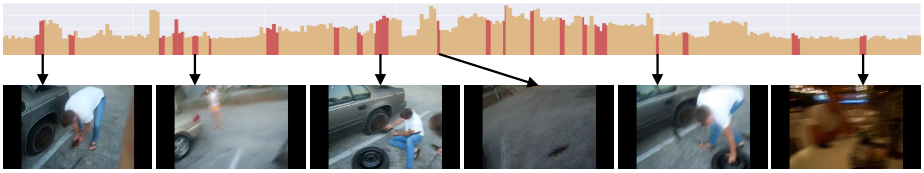
In this paper, we have proposed SumGraph to formulate video summarization as a graphical modeling problem. The key idea of our approach is to solve the problem of video summarization by constructing and recursively estimating a



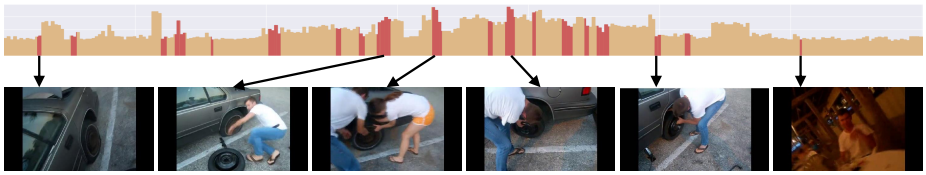
(a) SUM-FCN [30]



(b) SumGraph



(c) SUM-FCN [30]



(d) SumGraph

Fig. 6. Qualitative results on the TVSum benchmark [31]: (a) SUM-FCN [30], (b) SumGraph for video number 18, and (c) SUM-FCN [30], (d) SumGraph for video number 3. Brown bars show frame level user annotation. Red bars are selected subset shots. Best viewed in color.

relation graph to embed the frame-to-frame semantic interactions. With SumGraph, the obtained relation graph is exploited to infer a summary video based on semantic understanding of the whole frames in both supervised and unsupervised fashion. Our model showed significant improvement of performance over other approaches. We hope that the results of this study will facilitate further advances in video summarization and its related tasks.

Acknowledgement

This research was supported by R&D program for Advanced Integrated-intelligence for Identification (AIID) through the National Research Foundation of KOREA(NRF) funded by Ministry of Science and ICT (NRF-2018M3E3A1057289).

References

1. Open video project `https://open-video.org/`
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: TPAMI (2018)
3. De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., de Albuquerque Araújo, A.: Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Patt. Rec. Letters (2011)
4. Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., Remagnino, P.: Summarizing videos with attention. In: ACCVW (2018)
5. Gao, J., Zhang, T., Xu, C.: Graph convolutional tracking. In: CVPR (2019)
6. Gong, B., Chao, W.L., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: NeurIPS (2014)
7. Gygli, M., Grabner, H., Gool, L.V.: Video summarization by learning submodular mixtures of objectives. In: CVPR (2015)
8. Gygli, M., Grabner, H., Riemenschneider, H., Gool, L.V.: Creating summaries from user videos. In: ECCV (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016)
10. He, X., Hua, Y., Song, T., Zhang, Z., Xue, Z., Ma, R., Robertson, N., Guan, H.: Unsupervised video summarization with attentive conditional generative adversarial networks. In: ACMMM (2019)
11. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR (2016)
12. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-supervised learning with graph learning-convolutional networks. In: CVPR (2019)
13. Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., Cohen, M.F.: Real-time hyperlapse creation via optimal frame selection. ACM Trans. on Graphics (2015)
14. Kang, H.W., Matsushita, Y., Tang, X., Chen, X.Q.: Space-time video montage. In: CVPR (2006)
15. Kendall, M.G.: The treatment of ties in ranking problems. In: Biometrika **33**(3), 239–251 (1945)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
17. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS (2011)
18. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR (2012)
19. Liu, T., Kender, J.R.: Optimization algorithms for the selection of key frame sequences of variable length. In: ECCV (2002)
20. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: CVPR (2013)
21. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial lstm networks. In: CVPR (2017)
22. Ngo, C.W., Ma, Y.F., Zhang, H.J.: Automatic video summarization by graph modeling. In: ICCV (2003)
23. Nguyen, P., Liu, T., Prasad, G., Han, B.: Weakly supervised action localization by sparse temporal pooling network. In: CVPR (2018)

24. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J.: Rethinking the evaluation of video summaries. In: CVPR (2019)
25. Park, J., Lee, J., Jeon, S., Kim, S., Sohn, K.: Graph regularization network with semantic affinity for weakly-supervised temporal action localization. In: ICIP (2019)
26. Poleg, Y., Halperin, T., Arora, C., Peleg, S.: Egosampling: Fast-forward and stereo for egocentric videos. In: CVPR (2015)
27. Potapov, D., Douze, M., Harchaoui, Z., Schmid, C.: Category-specific video summarization. In: ECCV (2014)
28. Pritch, Y., Rav-Acha, A., Gutman, A., Peleg, S.: Webcam synopsis: Peeking around the world. In: ICCV (2007)
29. Rochan, M., Wang, Y.: Video summarization by learning from unpaired data. In: CVPR (2019)
30. Rochan, M., Ye, L., Wang, Y.: Video summarization using fully convolutional sequence networks. In: ECCV (2018)
31. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. In: CVPR (2015)
32. Sun, M., Farhadi, A., Taskar, B., Seitz, S.: Salient montages from unconstrained videos. In: ECCV (2014)
33. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
34. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
35. Wang, X., Gupta, A.: Videos as space-time region graphs. In: ECCV (2018)
36. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (2018)
37. Yuan, L., Tay, F.E., Li, P., Zhou, L., Feng, J.: Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. In: AAAI (2019)
38. Yuan, Y., Liang, X., Wang, X., Yeung, D.Y., Gupta, A.: Temporal dynamic graph lstm for action-driven video object detection. In: ICCV (2017)
39. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: ICCV (2019)
40. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Summary transfer: Examplar-based subset selection for video summarization. In: CVPR (2016)
41. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: ECCV (2016)
42. Zhang, K., Grauman, K., Sha, F.: Retrospective encoders for video summarization. In: ECCV (2018)
43. Zhao, B., Li, X., Lu, X.: Hierarchical recurrent neural network for video summarization. In: ACMMM (2017)
44. Zhou, K., Qiao, Y., Xiang, T.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: AAAI (2018)
45. Zwillinger, D., Kokoska, S.: Crc standard probability and statistics tables and formulae. In: CRC Press (1999)