A Memory Network Approach for Story-based Temporal Summarization of 360° Videos

Sangho Lee, Jinyoung Sung, Youngjae Yu, Gunhee Kim
Seoul National University
sangho.lee@vision.snu.ac.kr, jysung710@gmail.com, yj.yu@vision.snu.ac.kr, gunhee@snu.ac.kr
http://vision.snu.ac.kr/projects/pfmn

Abstract

We address the problem of story-based temporal summarization of long 360° videos. We propose a novel memory network model named Past-Future Memory Network (PFMN), in which we first compute the scores of 81 normal field of view (NFOV) region proposals cropped from the input 360° video, and then recover a latent, collective summary using the network with two external memories that store the embeddings of previously selected subshots and future candidate subshots. Our major contributions are twofold. First, our work is the first to address story-based temporal summarization of 360° videos. Second, our model is the first attempt to leverage memory networks for video summarization tasks. For evaluation, we perform three sets of experiments. First, we investigate the view selection capability of our model on the Pano2Vid dataset [42]. Second, we evaluate the temporal summarization with a newly collected 360° video dataset. Finally, we experiment our model's performance in another domain, with image-based storytelling VIST dataset [22]. We verify that our model achieves state-of-the-art performance on all the tasks.

1. Introduction

360° videos are growing rapidly as recording devices such as GoPro and GearVR spread widely and many social network platforms such as YouTube and Facebook eagerly support the sharing of this content. With the explosive growth of 360° content, there is a strong need for automatic summarization, despite that 360° video summarization has been still under-addressed in the video summarization literature. In particular, only spatial summarization, which controls *where* to look in the video frame of unlimited field of view (FOV) and generates an optimal camera trajectory, has been addressed in [42, 41, 21]. Recently, Yu *et al*. [52] attempt to generate a spatio-temporal highlight for a long 360° video, although they simply apply the ranking model for spatial summarization to the temporal domain without deliberate consideration of temporal summarization.

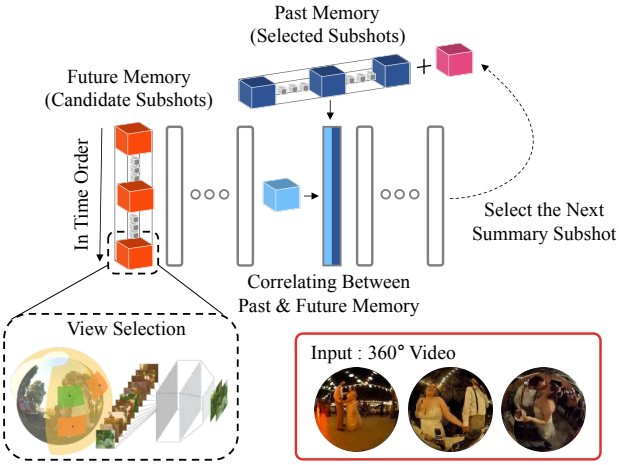

Figure 1. The intuition of the proposed PFMN model for temporal summarization of 360° videos. With a view selection module for finding key objects and a novel memory network leveraging two *past* and *future* memories, our PFMN temporally summarizes a 360° video based on the underlying storyline.

In this paper, we focus on story-based temporal summarization of long 360° videos by selecting a subset of key subshots. Specifically, we focus on tackling the following two summarization problems that are caused by the characteristics of 360° videos. First, 360° videos contain all surroundings of a camera in all directions. As a result, unlike normal videos, there is no specific subject that a videographer intends to shoot. Since identifying the subject is often crucial to grasp the plot of the entire video, its absence can be a stumbling block to the temporal summarization. Second, it is difficult to use supervised learning, which is preferable for model performance. There are few available pairs of long 360° videos and their corresponding edited summaries. To make matters worse, unlimited FOVs make it difficult to browse videos, thus it takes a lot of time and effort for humans to annotate the data.

To solve aforementioned problems, we propose a novel memory network model for 360° video temporal summarization, named as *Past-Future Memory Network* (PFMN), whose intuition is shown in Figure 1. As preprocessing, we first perform the view selection using a deep ranking network that is learned from photostream data of the same topic. That is, we compute the scores of 81 normal field of view (NFOV) region proposals cropped from the input 360° video. Then, we perform temporal summarization, with an assumption that the video set of the same topic shares the common *storyline*, which we define as an ordered sequence of subshot exemplars that frequently appear in the whole video set. We recover this latent, collective storyline, using a memory network involving two external memories that store the embeddings of previously selected subshots and future candidate subshots that are not selected yet. Based on the fact that humans make a video summary using the entire context, our model iteratively selects a next summary subshot using the correlation between the past and future information. Since 360° videos of good quality are not large-scale enough to train the proposed memory network, we pre-train the model using photostream data and fine-tune it using a small set of training 360° videos.

Temporal summarization is often formulated as sequence prediction, because it requires to choose a subsequence among the entire video. We design our model based on memory networks (*e.g*. [10, 11, 12, 27, 47]), instead of recurrent neural networks (RNN) [33] and their variants such as LSTM [19] and GRU [3], which may be one of the most dominant frameworks for sequence prediction. We argue that the memory network approach bears two major advantages. First, RNNs and their variants represent previous history with a hidden state of a fixed length, which may be often insufficient for processing a long video subshot sequence. On the other hand, the external memory significantly improves the model's memorization capability to explicitly store the whole visual information from the video input. Second, RNNs predict the very next item based on the hidden state only. However, the hidden state stores the information about the whole previous sequence, and some of this information, that is not relevant to the summary, may degrade the model performance. Yet, our model stores only the previously selected subshots, which helps our model focus on the storyline of the input video.

For evaluation, we perform three sets of experiments. First, we run spatial summarization experiments on the Pano2Vid dataset [42], to investigate the view selection capability of our model. Second, we evaluate story-based temporal summarization with a newly collected 360° video summarization dataset, which is the target task of this work. Finally, we evaluate our model's summarization performance in another domain, using an image-based storytelling VIST dataset [22]. We verify that our model outperforms the state-of-the-art methods on all the tasks.

Finally, we outline contributions of this work as follows.

1. To the best of our knowledge, our work is the first to address the problem of temporal summarization of 360° videos.
2. We propose a novel model named *Past-Future Memory Network* (PFMN). As far as we know, our model is the first attempt to leverage memory networks for the problem of video summarization, including not only 360° videos but also normal videos. The unique updates of PFMN include (i) exploiting an additional memory for future context, and (ii) correlating the future context and previously selected subshots to select the next summary subshot at each iteration.
3. We qualitatively show that our model outperforms state-of-the-art video summarization methods in various settings with not only our own 360° video dataset but also Pano2Vid [42] and VIST [22] benchmarks.

2. Related Work

Learning Storylines. Our model learns an underlying visual storyline that many videos of the same topic share. Some early methods for storyline inference require human expertise [38], while recent approaches make use of data-driven, unsupervised learning methods [1, 32, 46] or weakly-supervised ones [13, 48]. However, even for data-driven approaches, some pre-defined criteria like sparsity or diversity [26, 25] are needed for recovering storylines. One of the closest works to ours may be [39], which proposes Skipping Recurrent Neural Networks (S-RNN), to discover long-term temporal patterns in photostream data by skipping through images by adding *skipping* operations to classic RNNs. However, our model summarizes 360° videos rather than photostreams, and leverages the memory structure, instead of RNNs, to fully utilize long input sequences.

As another related task, visual storytelling is to generate coherent story sentences from an image set. Huang *et al*. [22] recently release the visual storytelling VIST dataset, and Yu *et al*. [51] propose a model to select representative photos and then generate story sentences from those summary photos. Since each album in the VIST has annotations to indicate which photos are used to generate stories, (*i.e*. which photos are representative), we also evaluate our summarization model on the VIST to demonstrate its performance in a different domain.

Temporal Video Summarization. [45] provides a thorough overview of earlier video summarization methods. Many approaches are unsupervised methods, but still use some heuristic criteria such as importance or representativeness [20, 28, 30, 24, 40], relevance [37, 4], and diversity or non-redundancy [29, 6, 5, 55]. There have been some supervised methods for temporal summarization, such as

[9, 14, 2, 15, 53]. Recently, a few approaches have leveraged deep neural architectures for video summarization. For unsupervised cases, Yang *et al*. [49] use robust recurrent auto-encoders to identify highlight segments, and Mahasseni *et al*. [31] propose a video summarization approach based on variational recurrent auto-encoders and generative adversarial networks. For supervised cases, Zhang *et al*. [54] integrate LSTMs with the determinantal point process for video summarization. Deep pairwise ranking models have been also popularly used for video highlight detection such as [50, 16]. However, unlike ours, no previous work is based on the memory network framework. Our work can be categorized as a weakly supervised approach that uses image priors for video summarization (*e.g*. [40, 24, 25]), although we directly use raw photostreams with no summary annotation. Moreover, our work is the first to attempt 360° video temporal summarization.

360° Video Summarization. Despite the explosive increase of 360° video content, its summarization has been still understudied, except the *AutoCam* framework [41, 42], *deep 360 pilot* [21], and *Composition View Score* (CVS) model [52]. AutoCam tackles the Pano2Vid problem, which takes a 360° video and generates NFOV camera trajectories as if a human videographer would take with a normal NFOV camera (*i.e*. the spatial summarization of 360° videos). The CVS framework proposes a deep ranking model for spatial summarization to select NFOV shots from each frame of a 360° video, and extends the same model for the temporal domain to generate a spatio-temporal highlight video. However, it is hard to say that the CVS framework is designed considering the issues of temporal summarization. Moreover, the objective of our work is different from those of previous works in that we produce a concise abstraction of a 360° video that recovers the underlying visual storyline, rather than assessing the importance of video segments.

3. Datasets

We collect a dataset of 360° videos for temporal summarization, along with photostream data with the same topics. We exploit the photostream data for two purposes: (i) training the 360° video view selection module, and (ii) pretraining our memory network model for temporal summarization. Their key statistics are summarized in Table 1-2.

3.1. 360° Video Dataset

We newly collect 360° videos from *YouTube* with five query terms: *wedding*, *parade*, *rowing*, *scuba diving*, and *airballooning*. All the topics have a fairly consistent narrative structure, and involve a large volume of both 360° videos and photostreams. For each topic, we download as many videos as possible, and manually filter out those that are irrelevant to the topic. Since these videos are used for temporal summarization, we also ignore the ones that are less than five minutes long. In total, we obtain 285 videos with a combined duration of about 92 hours.

Topics	360° Videos		
	# videos	total (hr)	mean (min)
wedding	50	27.52	33.03
parade	82	33.18	24.27
rowing	41	9.25	13.53
scuba diving	73	13.45	11.05
airballooning	39	8.83	13.58

Table 1. Statistics on our new 360° video summarization dataset.

Topics	Photostreams (PS)		
	# PS	# imgs	# imgs per PS
wedding	1,071	216,768	205
parade	718	110,789	154
rowing	2,019	166,335	82
scuba diving	2,735	176,461	65
airballooning	2,595	176,093	68

Table 2. Statistics on our photostream dataset.

3.2. Photostream Dataset

We gather collections of Flickr photostreams from the YFCC100M dataset [44] for two topics, *wedding* and *parade*, and the dataset of Kim and Xing [26] for the other topics. We select only photostreams that include more than 100 photos taken on the same day, and then sort each photostream based on the photos' time taken. In total, the dataset consists of about 85K images of 9K photostreams.

3.3. Video Formats

A 360° video frame cannot be projected in a 2D plane with no distortion. Thus, we first select a viewpoint in a form of longitude and latitude coordinates (ϕ, θ) in the spherical coordinate system, and extract an NFOV region from the 360° frame by a rectilinear projection with the viewpoint as center. The ideal size of an NFOV region may need to vary according to several factors (*e.g*. the sizes of key objects or the geometry of filming environment). We heuristically set the size of an NFOV region to span 54° horizontally and 30° vertically, while fixing the aspect ratio to 16:9, because it performs the best in experiments.

If some distortion is allowed, a whole spherical frame can be represented by equirectangular projection (ERP), which transforms the spherical coordinates of a 360° frame into the planar coordinates as a 2D world map from the spherical Earth. The resulting x, y coordinates in the ERP format are: $x = (\phi - \phi_0)\cos\theta_1, y = (\theta - \theta_1)$, where ϕ_0 is the central meridian and θ_1 are the standard parallels (north and south of the equator) in the spherical coordinate system.

4. Past-Future Memory Network (PFMN)

Figure 2 shows the overall architecture of *Past-Future Memory Network* (PFMN). The input to the model is a sequence of video subshots $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ for an input 360° video sampled at 5 fps. We construct $\mathcal{V}$ using the

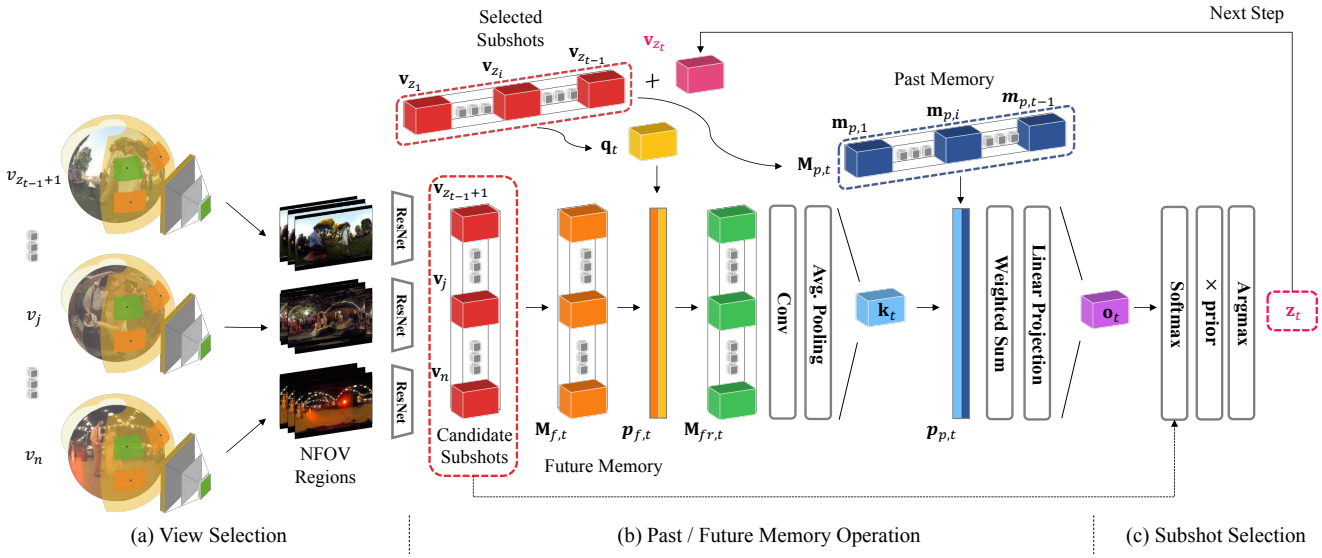

Figure 2. Architecture of the proposed *Past-Future Memory Network*. (a) We compute the key-objectness scores of NFOV regions from each spherical subshot using a deep ranking network (section 4.1). (b) The past memory $\mathbf{M}_p$, which stores already selected subshots, and the future memory $\mathbf{M}_f$, which saves the remaining subshots, are correlated via convolutional operations and soft attention models to compute output $\mathbf{o}_t$ (section 4.2). (c) Finally, the subshot from the future memory with the highest score is chosen as the next subshot summary, and move into the past memory for next iteration (section 4.2).

Kernel Temporal Segmentation (KTS) [37], maintaining visual coherence within each subshot. Subshots have an average duration of 6.1s. The output is a selected subsequence $\mathcal{S} = \{v_{z_1}, v_{z_2}, \cdots, v_{z_m}\} \subset \mathcal{V}$ as a summary, where m is a user parameter to set the length of the summary.

We first run a preprocessing step that extracts a set of 81 NFOV candidates with key-objectness scores from each 360° subshot v_i (section 4.1). We then run temporal summarization using our proposed PFMN model (section 4.2). We design the view selection module by a simple modification of standard deep ranking models, and our major technical novelties lie in the memory network part of PFMN.

4.1. 360° Video View Selection

The goal of the 360° video view selection is to assign key objectness scores to a set of NFOV candidates from each subshot v_i of a whole spherical frame. We sample 81 NFOV candidate regions at longitudes $\phi \in \Phi = \{0°, 40°, 80°, \cdots, 320°\}$ and latitudes $\theta \in \Theta = \{0°, \pm 15°, \pm 35°, \pm 55°, \pm 75°\}$ from the middle frame of v_i, as depicted in Figure 3. We use a trained deep ranking model for computing the key-objectness score of each candidate region $v_{i,j}$ for $j = 1, 2, \cdots, 81$. Specifically, we define the key object as an object that people are likely to keep track of when they shoot normal videos of the same topic: for example, the bride and groom in wedding videos and tropical fishes in scuba diving videos. Since 360° videos record all surroundings of a camera in all directions, we want to filter out irrelevant objects and only keep the key objects as the input to a 360° video summarization model.

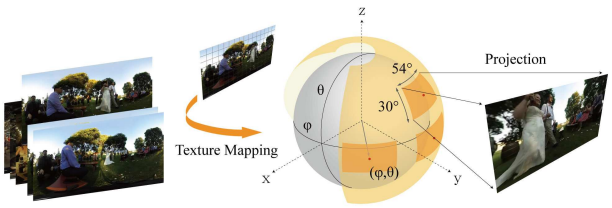

Figure 3. NFOV projection. We extract an NFOV region from a 360° frame by a rectilinear projection with a specified viewpoint as center. The size of each NFOV region spans a horizontal angle 54° and a vertical angle 30° for a 16:9 aspect ratio.

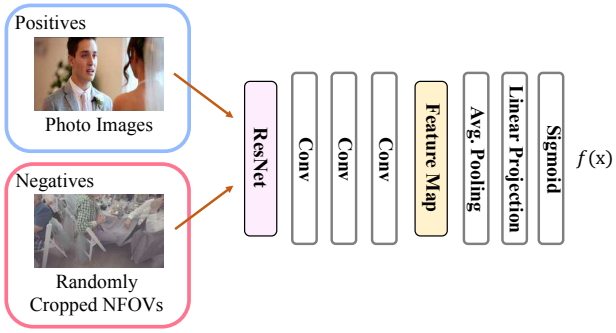

Figure 4. The architecture of our view selection model. Unlike RankNet [16], our ranking model consists of convolutional layers followed by a global average-pooling layer, a linear projection layer and a sigmoid activation. We use stride 1 and no zero-padding for each convolutional layer.

Model. Inspired by RankNet [16], we leverage a deep ranking network for computing the key-objectness score $f(\mathbf{x})$ of an input NFOV $\mathbf{x}$ (See Figure 4). While RankNet

consists of fully-connected layers, our ranking model is based on convolutional (Conv) layers to exploit the spatial attention. We represent the input using the res5c feature map $\mathbf{x}^{r5c} \in \mathbb{R}^{7\times7\times2048}$ of ResNet-101 [18], pretrained on ImageNet [7]. We then feed it into three Conv layers with filters $\mathbf{w}^i_{rank} \in \mathbb{R}^{2\times2\times c^{i-1}_{rank}\times c^i_{rank}}$ for $i = 1, 2, 3$:

$$\mathbf{x}^i_{rank} = \text{ReLU}(\text{conv}(\mathbf{x}^{i-1}_{rank}, \mathbf{w}^i_{rank})), \quad (1)$$

where conv (input, filter) indicates the Conv layer, and ReLU is an element-wise ReLU activation [35]. We set $\mathbf{x}^0_{rank} = \mathbf{x}^{r5c}$ and $c^{i=\{1,2,3\}}_{rank} = \{2048, 1024, 512\}$ as filter channels. We apply Batch Normalization [23] to every Conv layer. Finally, we apply a global average pooling, a linear projection, and a sigmoid activation to $\mathbf{x}^3_{rank} \in \mathbb{R}^{5\times5\times512}$, to compute key-objectness score $f(\mathbf{x})$.

Training. We regard photo images as positives, and randomly cropped NFOV regions as negatives. That is, we define the ranking constraint over the training set: $f(p_i) \succ f(c_i), \forall\ (p_i, c_i) \in \mathcal{D}_{rank}$, where p_i and n_i are positive and negative samples, respectively. We then train our ranking model using the max-margin loss $\mathcal{L}_{rank,i} = \max(0, f(n_i) - f(p_i) + 1)$. The final objective is the total loss over $\mathcal{D}_{rank}$ with l_2 regularization:

$$\mathcal{L}_{rank} = \sum_i \mathcal{L}_{rank,i} + \lambda||\mathcal{M}_{rank}||^2_F, \quad (2)$$

where $\mathcal{M}_{rank}$ denotes the model parameters and λ is a regularization hyperparameter.

4.2. Story-based Temporal Summarization

When humans summarize a video, they may first watch the whole video to understand the entire context, and then choose one by one by comparing between what they have already selected and what remains in the video. Based on this intuition, our temporal summarization includes two external memories: *past memory* and *future memory*, as one of our key novelties. The past memory $\mathbf{M}_p$ stores the already selected subshots as summary, while the future memory $\mathbf{M}_f$ stores the remaining subshots of the video. Initially, the past memory is empty (*i.e.* $\mathbf{M}^{(0)}_p = \emptyset$), and the future memory includes all the subshots of the video (*i.e.* $\mathbf{M}^{(0)}_f = \mathcal{V}$). At each iteration t, we select v_{z_t} from the future memory as the best (*i.e.* the most plausible) subshot to be included in the summary, and then move v_{z_t} to the past memory. At next iteration $t+1$, the future memory becomes $\mathbf{M}^{(t+1)}_f = \{v_{z_t+1}, \cdots, v_n\}$; we only allow the subshots after the latest summary subshot as candidates to be chosen. We iterate this process m times; eventually the summary is identical to the past memory elements: $\mathcal{S} = \{v_{z_1}, v_{z_2}, \cdots, v_{z_m}\} = \mathbf{M}^{(m)}_p$. We assume here that the summary subshots are sorted in a chronological order.

Subshot Representation. We convert each subshot v_i into feature representation as follows. For each of 81 NFOV candidates $\{v_{i,j}\}^{81}_{j=1}$ from v_i, we extract the pool5 feature vector $\mathbf{v}_{i,j}$ of ResNet-101 [18] pre-trained on ImageNet. We then sum over all regions as $\mathbf{v}_i = \sum_j w_j \mathbf{v}_{i,j} \in \mathbb{R}^{2,048}$, where w_j is the normalized key-objectness score of $v_{i,j}$ that the view selection model computes.

Past and Future Memory. At iteration t, the past memory stores previous summary subshots $\{v_{z_1}, \cdots, v_{z_{t-1}}\}$. Following the standard representation of memory slots (*e.g.* [43]), we encode each of them into input and output embeddings using different parameters $\mathbf{W}^{i/o}_p \in \mathbb{R}^{1024\times2048}$ and $\mathbf{b}^{i/o}_p \in \mathbb{R}^{1024}$, where i and o stand for input and output embeddings, respectively. The resulting past memory vector $\mathbf{m}^{i/o}_{p,j} \in \mathbb{R}^{1024}$ is represented by

$$\mathbf{m}^{i/o}_{p,j} = \text{ReLU}(\mathbf{W}^{i/o}_p \mathbf{v}_{z_j} + \mathbf{b}^{i/o}_p), \quad j = 1, \cdots, z_{t-1}. \quad (3)$$

The future memory stores the information about future subshots, $\{v_{z_{t-1}+1}, \cdots, v_n\}$. Similar to the past memory, the future memory vector $\mathbf{m}^{i/o}_{f,j} \in \mathbb{R}^{1024}$ is represented as input and output embeddings:

$$\mathbf{m}^{i/o}_{f,j} = \text{ReLU}(\mathbf{W}^{i/o}_f \mathbf{v}_j + \mathbf{b}^{i/o}_f), \quad j = z_{t-1}+1, \cdots, n \quad (4)$$

where $\mathbf{W}^{i/o}_f \in \mathbb{R}^{1024\times2048}$ and $\mathbf{b}^{i/o}_f \in \mathbb{R}^{1024}$.

Finally, we stack the memory vectors row by row for later computation: $\mathbf{M}^{i/o}_{p,t} = [(\mathbf{m}^{i/o}_{p,1})^T; \cdots; (\mathbf{m}^{i/o}_{p,t-1})^T]$ and $\mathbf{M}^{i/o}_{f,t} = [(\mathbf{m}^{i/o}_{f,z_{t-1}+1})^T; \cdots; (\mathbf{m}^{i/o}_{f,n})^T]$.

Correlating between Past and Future Memory. Rather than using recurrent connections throughout time as RNNs do, our model predicts the next summary subshot by extracting a key vector from the future memory, and computing the attention of the past memory with respect to the key vector. Inspired by RWMN [34], we use convolutions with a learnable kernel as a read operation on the memory. We compute a query embedding $\mathbf{q}_t$ from the mean-pooled summary $\mathbf{v}_{avg} = (\sum^{t-1}_{j=1} \mathbf{v}_{z_j})/(t-1)$:

$$\mathbf{q}_t = \text{ReLU}(\mathbf{W}_q \mathbf{v}_{avg} + \mathbf{b}_q), \quad (5)$$

where $\mathbf{W}_q \in \mathbb{R}^{1024\times2048}$ and $\mathbf{b}_q \in \mathbb{R}^{1024}$. $\mathbf{q}_t$ can be interpreted as an indicator of the current story flow based on the previously selected summary subshots $\{v_{z_1}, \cdots, v_{z_{t-1}}\}$. We intend to dynamically update the future memory embedding according to $\mathbf{q}_t$ at every step. Based on this intuition, $\mathbf{q}_t$ is fed into the soft attention model of the future memory:

$$\mathbf{p}_{f,t} = \text{softmax}(\mathbf{M}^i_{f,t} \mathbf{q}_t), \quad (6)$$

$$\mathbf{M}_{fr,t}[i,:] = \mathbf{p}_{f,t}[i] \mathbf{M}_{f,t}[i,:], \quad (7)$$

where $\mathbf{p}_{f,t} \in \mathbb{R}^{n-z_{t-1}}$ and $\mathbf{M}_{fr,t} \in \mathbb{R}^{(n-z_{t-1})\times1024}$. It means that we compute how well the query embedding $\mathbf{q}_t$

is compatible with each cell of future memory by softmax on the inner product (Eq.(6)), and rescale each cell by the element-wise multiplication with the attention vector $\mathbf{p}_{f,t}$ (Eq.(7)). We regard the attended output memory $\mathbf{M}_{fr,t}$ as the summarization context.

Since adjacent subshots in videos connecting storylines often have strong correlations with one another, we associate neighboring memory cells of the attended memory by applying a Conv layer to $\mathbf{M}_{fr,t}$. The Conv layer consists of a filter $\mathbf{w}_{fr} \in \mathbb{R}^{k_v \times k_h \times 1024 \times 1024}$, whose vertical and horizontal filter size and strides are $k_v = 20, k_h = 1024, s_v = 10, s_h = 1$, respectively. We then apply an average-over-time pooling to obtain the key vector $\mathbf{k}_t \in \mathbb{R}^{1024}$, which can be regarded as a concise abstraction of future context with respect to the current query $\mathbf{q}_t$:

$$\mathbf{k}_t = \text{averagepool}(\text{conv}(\mathbf{M}_{fr,t}, \mathbf{w}_{fr}, \mathbf{b}_{fr})), \quad (8)$$

where $\mathbf{b}_{fr} \in \mathbb{R}^{1024}$ is a bias vector.

We then compute the soft attention of the past memory through the inner product between $\mathbf{k}_t$ and each cell of $\mathbf{M}^i_{p,t}$. The final memory output $\mathbf{m}_t \in \mathbb{R}^{1024}$ becomes an abstraction of the past memory $\mathbf{M}_{p,t}$ using the future context $\mathbf{k}_t$:

$$\mathbf{m}_t = \mathbf{p}^T_{p,t}\mathbf{M}^o_{p,t}, \quad \mathbf{p}_{p,t} = \text{softmax}(\mathbf{M}^i_{p,t}\mathbf{k}_t), \quad (9)$$

where $\mathbf{p}_{p,t} \in \mathbb{R}^{t-1}$. This part is the contact point where past and future information are combined, and is the primary novelty of our model.

Selecting the Next Summary. Finally, we select the next summary using the final memory output $\mathbf{m}_t$. By using a linear projection, we extend the dimension of $\mathbf{m}_t$ to the output $\mathbf{o}_t \in \mathbb{R}^{2048}$, and compute the compatibility scores c_j between $\mathbf{o}_t$ and each future subshot $\mathbf{v}_j$ for $j = z_{t-1} + 1, \cdots, n$, using the inner product and the softmax operation: $c_j = \text{softmax}(\mathbf{o}^{'}_t\mathbf{v}_j)$.

We then multiply c_j by a prior $u_{j,t}$ to be the selection probability $s_j = c_j u_{j,t}$. The prior is the probabilistic equivalent of random sampling without replacement, considering the selected subshot sequence is an ordered subset:

$$u_{j,t} = \begin{cases} \prod_{k=z_{t-1}+1}^{j-1}(1-u_{k,t})\frac{m-t+1}{n-t+1} & j \leq n-m+t, \\ 0 & j > n-m+t. \end{cases}$$

This prior simply conveys two obvious constraints: (i) the farther away from the subshot selected in the previous iteration, the lower the probability is assigned, and (ii) the probability of a subshot that should not be selected at the current iteration is zero (*e.g.* when $n = 10, m = 4$, it prevents selecting the ninth subshot as the second summary).

Finally, we select the next summary v_{z_t} in a greedy way; that is, we select the subshot with the highest selection probability: $z_t = \text{argmax}_{j \in \{z_{t-1}+1, \cdots, n\}} s_j$. Until the number of selected subshots reaches a user-defined m, the selected summary v_{z_t} moves back into the past memory, and we repeat all the above selection process again.

4.3. Training

The goal of training is to learn transitions between summary subshots, so as to recover the latent storyline and summarize individual videos. Since we have no annotation for true key subshots in the training set, we optimize the likelihood of a selected subshot at each iteration t (*i.e.* maximize the softmax probability c_{z_t}). This is mathematically based on the training procedure of S-RNN [39], which ensures that the model parameters converge to a local optimum. Since 360° video training data are not large-scale, we use both photostream data and 360° video data for training. We first train the parameters of the memory network using photostreams, and then fine-tune using the 360° video training set. Note that our method does not require groundtruth summary for both photostreams and 360° videos, and instead uses the raw photostreams and videos for training.

Thus, our loss function over the training set $\mathcal{D}$ is

$$\mathcal{L} = \sum_{\mathcal{V} \in \mathcal{D}} \sum_{j=1}^{m} -c_{z_j}, \quad (10)$$

where $\mathcal{V}$ is a photostream for pre-training and a 360° video for fine-tuning. To make the view selection more optimized for summarization, we fine-tune the parameters of the view selection model along with those of the memory network.

Implementation Details. We initialize all the parameters via random sampling from a Gaussian distribution with standard deviation of $\sqrt{2/\text{dim}}$, following He *et al.* [17]. For optimizing the objective of the view selection in Eq.(2), we use a SGD with a mini-batch size of 16, a Nesterov momentum of 0.5, and λ to $1e-07$. We set our initial learning rate as 0.0001 and divide it by 2 at every 16 epochs. For optimizing the objective of the memory network in Eq.(10), we select the AdaGrad [8] optimizer with a a learning rate of 0.001 and an initial accumulator value of 0.1.

5. Experiments

We evaluate the PFMN from three aspects. First, we run spatial summarization experiments on the Pano2Vid dataset [42], to study the view selection of our model. Second, we evaluate the story-based temporal summarization on our newly collected 360° video dataset, on which our model achieves the state-of-the-art performance. Finally, we evaluate our model's performance in another domain; using the image-based storytelling dataset (VIST) [22].

5.1. Experimental Setting

View Selection. As a proxy study for capturing key objects, we measure the performance of spatial summarization on the Pano2Vid dataset, consisting of 86 360° videos of four topics. The center coordinates of the selected regions are annotated in the spherical coordinates (*i.e.* latitude and longitude) at each segment. Using them as

Methods	Frame cosine sim	Frame overlap
Center [42]	0.572	0.336
Eye-Level [42]	0.575	0.392
Saliency [42]	0.387	0.188
AutoCam [42]	0.541	0.354
AutoCam-stitch [42]	0.581	0.389
TS-DCNN [50]	0.578	0.441
RankNet [16]	0.562	0.398
PFMN	**0.661**	**0.536**

Table 3. Experimental results of spatial summarization on the Pano2Vid [42] dataset. Higher values are better in both metrics.

groundtruth (GT), we compare the similarity between the human-selected trajectories and algorithm-generated ones. We use the metrics of mean cosine similarity and mean overlap as in the Pano2Vid benchmark [42].

Temporal Summarization. For evaluation, we randomly sample 10 360° videos per topic as a test set, and then obtain three GT summaries per video from human annotators. To avoid the chronological bias [40], a tendency that humans prefer the shots that appear earlier in videos, we construct the GTs of key subshots as follows. For each test video, we ask an annotator to watch the whole video and mark all the frames that they think are important in the story flow. We then segment the video into subshots using the Kernel Temporal Segmentation (KTS) [37], and rank the subshots in the descending order by f_{is}/f_i, where f_i is the number of frames in subshot i, and f_{is} is the number of selected frames. For annotation, we let the total duration of GT to be below 15% of the whole video. We recruit five annotators with different backgrounds. Following video summarization literature, we use the average pairwise F_1-measure compared to GT as evaluation metrics.

Storyline Evaluation. The VIST dataset consists of 10,000 Flickr photo albums with 200,000 images for 69 topics. Each album includes 10 to 50 photos taken within a 48-hour span with two GT summaries, each consisting of five human-selected photos. We select twelve topics with the largest number of albums. We set aside 10% of the training set for validation. We compute the precision and recall of generated summaries using the combined set of GT stories.

5.2. Baselines

For view selection (*i.e.* spatial summarization), we use six baselines: (i) three simple baselines of spatial summarization (Center, Eye-level and Saliency) used in [42], (ii) AutoCam [42], and (iii) two state-of-the-art pairwise deep ranking models for highlight detection, TS-DCNN [50] and RankNet [16]. Since deep 360 pilot [21] requires labeled data for training, it is not selected as our baseline.

For temporal summarization, we use five unsupervised algorithms (random/uniform sampling, VSUMM [6, 5], SUM-GAN [31], and S-RNN [39]) and two deep ranking algorithms (TS-DCNN and RankNet). The random/uniform sampling, VSUMM, and SUM-GAN are keyframe based methods. Thus, we transform the automatically selected frames into key subshots as done in constructing subshot GTs in section 5.1. For TS-DCNN and RankNet, we use photostream and randomly selected video subshots as positive and negative samples, respectively.

Methods	F_1-measure (%)
random sampling	13.01
uniform sampling	14.60
VSUMM [6, 5]	12.38
SUM-GAN [31]	17.72
TS-DCNN [50]	16.24
RankNet [16]	15.96
S-RNN [39]	19.59
PFMN-FF	23.15
PFMN-FA	24.40
PFMN-PA	24.14
PFMN-noview	23.28
PFMN-RankNet	23.69
PFMN-hard	24.27
PFMN	**24.60**

Table 4. Experimental results of temporal summarization on our 360° video dataset.

Methods	Precision (%)	Recall (%)
K-means	43.76	27.71
S-RNN [39]	45.17	28.57
S-RNN- [39]	39.91	25.22
h-attn-rank [51]	45.30	28.90
PFMN	**50.42**	**31.85**

Table 5. Experimental results of storyline evaluation on the image-based (VIST) [22] dataset.

As an ablation study, we test six variants of our method. First, we use three different configurations of past and future memory: (i) the future memory with only 5% of the remaining subshots (PFMN-FF) or (ii) the whole subshots in the entire video (PFMN-FA), and (iii) the past memory with all the subshots that are not in the future memory (PFMN-PA). Furthermore, we test three variants with different view selection methods: (iv) with no view selection (PFMN-noview), (v) view selection with RankNet (PFMN-RankNet) and (vi) hard view selection (PFMN-hard) that chooses only top-K NFOV regions. We set K to 12.

For storyline evaluation, we use four baselines: K-means clustering, S-RNN, S-RNN- and a hierarchically-attentive RNN with ranking regularization (h-attn-rank) [51].

We implement the baselines as follows. For h-attn-rank and VSUMM, we use the code by the authors. For AutoCam and their baselines (Center, Eye-level and Saliency), we simply report the numbers in the original paper. We implement the other baselines using PyTorch [36] by ourselves.

5.3. Quantitative Results

View Selection. Table 3 compares the performance of our PFMN with those of baselines for spatial summarization on the Pano2Vid dataset. PFMN outperforms all the

Figure 5. Qualitative examples of 360° video summaries generated by S-RNN [39] and our PFMN, along with groundtruth (GT) summaries. Each subshot is represented by a sampled frame in the ERP format. Red boxes indicate the matches with the GT and prediction.

baselines by substantial margins in terms of both frame cosine similarity and frame overlap. It implies that our model captures key objects well in 360° views. Except the effect of the smooth-motion constraint (AutoCam-stitch), which enforces that longitude and latitude differences between consecutive segments must be less or equal than 30° so as to produce better camera trajectories ($|\phi_t - \phi_{t-1}|, |\theta_t - \theta_{t-1}| \leq 30°$), all the ranking models (RankNet, TS-DCNN and PFMN) achieve better results than the binary classification model (AutoCam). This suggests that formulating the spatial summarization as a ranking problem is more appropriate than as a binary classification problem of whether a view is in good composition or not. With the smoothing constraint, our PFMN even outperforms the improved version of AutoCam [41], which allows three FOVs per segment for view selection rather than a single fixed FOV (PFMN: 0.641 vs. [41]: 0.630 in terms of the frame overlap).

Temporal Summarization. Table 4 shows the results of temporal summarization on our new 360° video dataset. Our PFMN achieves better results compared to baselines with large margins. Among our variants, the PFMN with only the next few subshots leads to a performance drop (PFMN-FF: -1.45). The performance also slightly decreases when we use the whole video subshots for the future memory (PFMN-FA: -0.20). This may be due to the information redundancy with the past memory. Too much information in the past memory also drops the performance (PFMN-PA: -0.46), which implies that unnecessary subshots in the past memory may distract the selection of the next-best summary subshot. We also validate that the quality of view selection is critical for the performance of 360° video summarization (PFMN-noview: -1.32, PFMN-RankNet: -0.91). The small gap between hard and soft view selection (PFMN-hard: -0.33) indicates that only a small region of each 360° frame contains important objects.

Storyline Evaluation. Table 5 shows the results of storyline evaluation on the image-based VIST dataset. For fair comparison, we use the same feature representation, the pool5 feature of ResNet-101 [18], for all models. Our PFMN has higher performance compared to all the baselines, suggesting that our model also learns the latent storylines from images more successfully than baseline models, although our method is designed for 360° videos. It is notable that our model even achieves better results than h-attn-rank, which is a supervised method.

5.4. Qualitative Results

Figure 5 shows some qualitative examples of temporal summarization on 360° test videos of our dataset. In 360° subshots in the ERP (equirectangular projection) format, we show the matched subshots between the prediction and GT summaries in red boxes. The proposed PFMN not only recovers much of the GT summaries labeled by human annotators, but also successfully captures the main events in the storyline of the test video.

6. Conclusion

We proposed the *Past-Future Memory Network* model for story-based temporal summarization of 360° videos. Our model recovered latent, collective storyline using a memory network that involves two external memories to store the embeddings of previously selected subshots and future candidate subshots. We empirically validated that the proposed memory network approach outperformed other state-of-the-art methods, not only for view selection but also for story-based temporal summarization in both 360° videos and photostreams. We believe that there are several future research directions that go beyond this work. First, we can apply our approach to other types of summarization in the domains of text or images, because the proposed memory network has no limitation on the data modality. Second, we can extend our method to be a purely unsupervised method that automatically detects the storylines from only 360° video corpus without aid of photostreams.

Acknowledgements. This work was supported by the Visual Display Business (RAK0117ZZ-21RF) of Samsung Electronics, and IITP grant funded by the Korea government (MSIT) (No. 2017-0-01772). Gunhee Kim is the corresponding author.

References

[1] N. Chambers and D. Jurafsky. Unsupervised Learning of Narrative Event Chains. In *ACL*, 2008.

[2] W.-L. Chao, B. Gong, K. Grauman, and F. Sha. Large-Margin Determinantal Point Processes. In *UAI*, 2015.

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, 2014.

[4] W.-S. Chu, Y. Song, and A. Jaimes. Video Co-summarization: Video Summarization by Visual Co-occurrence. In *CVPR*, 2015.

[5] S. S. de Almeida, A. C. de Nazaré Júnior, A. de Albuquerque Arauújo, G. Cámara-Chávez, and D. Menotti. Speeding up a Video Summarization Approach Using GPUs and Multicore CPUs. *Procedia Computer Science*, 29:159–171, 2014.

[6] S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. 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.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[8] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, pages 2121–2159, 2011.

[9] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse Sequential Subset Selection for Supervised Video Summarization. In *NIPS*, 2014.

[10] A. Graves, G. Wayne, and I. Danihelka. Neural Turing Machines. *arXiv preprint arXiv:1410.5401*, 2014.

[11] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature*, 538:471–476, 2016.

[12] C. Gulcehre, S. Chandar, K. Cho, and Y. Bengio. Dynamic Neural Turing Machine with Soft and Hard Addressing Schemes. In *ICLR*, 2017.

[13] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding Videos, Constructing Plots Learning a Visually Grounded Storyline Model from Annotated Videos. In *CVPR*, 2009.

[14] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating Summaries from User Videos. In *ECCV*, 2014.

[15] M. Gygli, H. Grabner, and L. Van Gool. Video Summarization by Learning Submodular Mixtures of Objectives. In *CVPR*, 2015.

[16] M. Gygli, Y. Song, and L. Cao. Video2GIF: Automatic Generation of Animated GIFs from Video. In *CVPR*, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv preprint arXiv:1502.01852*, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[19] S. Hochreiter and J. Schmidhuber. Long Short-term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] R. Hong, J. Tang, H.-K. Tan, S. Yan, C. Ngo, and T.-S. Chua. Event Driven Summarization for Web Videos. In *SIGMM Workshop*, 2009.

[21] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 Pilot: Learning a Deep Agent for Piloting Through 360° Sports Videos. In *CVPR*, 2017.

[22] T.-H. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, J. Devlin, A. Agrawal, R. Girshick, X. He, P. Kohli, D. Batra, et al. Visual Storytelling. In *NAACL*, 2016.

[23] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

[24] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. In *CVPR*, 2013.

[25] G. Kim, L. Sigal, and E. P. Xing. Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. In *CVPR*, 2014.

[26] G. Kim and E. P. Xing. Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos. In *CVPR*, 2014.

[27] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *ICML*, 2016.

[28] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering Important People and Objects for Egocentric Video Summarization. In *CVPR*, 2012.

[29] Y. Li and B. Merialdo. Multi-Video Summarization Based on Video-MMR. In *WIAMIS Workshop*, 2010.

[30] Z. Lu and K. Grauman. Story-Driven Summarization for Egocentric Video. In *CVPR*, 2013.

[31] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised Video Summarization with Adversarial LSTM Networks. In *CVPR*, 2017.

[32] N. McIntyre and M. Lapata. Learning to Tell Tales: A Data-driven Approach to Story Generation. In *ACL*, 2009.

[33] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent Neural Network Based Language Model. In *Interspeech*, 2010.

[34] S. Na, S. Lee, J. Kim, and G. Kim. A Read-Write Memory Network for Movie Story Understanding. In *ICCV*, 2017.

[35] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 2010.

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Workshop*, 2017.

[37] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-Specific Video Summarization. In *ECCV*, 2014.

[38] R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Psychology Press, 2013.

[39] G. A. Sigurdsson, X. Chen, and A. Gupta. Learning Visual Storylines with Skipping Recurrent Neural Networks. In *ECCV*, 2016.

[40] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. TVSum: Summarizing Web Videos Using Titles. In *CVPR*, 2015.

[41] Y.-C. Su and K. Grauman. Making 360° Video Watchable in 2D: Learning Videography for Click Free Viewing. In *CVPR*, 2017.

[42] Y.-C. Su, D. Jayaraman, and K. Grauman. Pano2Vid: Automatic Cinematography for Watching 360° Videos. In *ACCV*, 2016.

[43] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-End Memory Networks. In *NIPS*, 2015.

[44] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The New Data in Multimedia Research. *arXiv preprint arXiv:1503.01817*, 2015.

[45] B. T. Truong and S. Venkatesh. Video Abstraction: A Systematic Review and Classification. *ACM TOMM*, 3(1):3, 2007.

[46] D. Wang, T. Li, and M. Ogihara. Generating Pictorial Storylines via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *AAAI*, 2012.

[47] J. Weston, S. Chopra, and A. Bordes. Memory Networks. *ICLR*, 2015.

[48] B. Xiong, G. Kim, and L. Sigal. Storyline Representation of Egocentric Videos with an Application to Story-based Search. In *ICCV*, 2015.

[49] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised Extraction of Video Highlights Via Robust Recurrent Auto-encoders. In *CVPR*, 2015.

[50] T. Yao, T. Mei, and Y. Rui. Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization. In *CVPR*, 2016.

[51] L. Yu, M. Bansal, and T. L. Berg. Hierarchically-Attentive RNN for Album Summarization and Storytelling. In *EMNLP*, 2017.

[52] Y. Yu, S. Lee, J. Na, J. Kang, and G. Kim. A Deep Ranking Model for Spatio-Temporal Highlight Detection from a 360° Video. In *AAAI*, 2018.

[53] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary Transfer: Exemplar-based Subset Selection for Video Summarization. In *CVPR*, 2016.

[54] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video Summarization with Long Short-term Memory. In *ECCV*, 2016.

[55] B. Zhao and E. P. Xing. Quasi Real-Time Summarization for Consumer Videos. In *CVPR*, 2014.