Unsupervised Video Summarization with Attentive Conditional Generative Adversarial Networks

He, X., Hua, Y., Song, T., Zhang, Z., Xue, Z., Ma, R., Robertson, N., & Guan, H. (2019). Unsupervised Video Summarization with Attentive Conditional Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 2296–2304). Association for Computing Machinery. https://doi.org/10.1145/3343031.3351056

Published in:
Proceedings of the 27th ACM International Conference on Multimedia

Document Version:
Peer reviewed version

Unsupervised Video Summarization with Attentive Conditional Generative Adversarial Networks

Xufeng He [1], Yang Hua[2], Tao Song[1,*]
Zongpu Zhang[1], Zhengui Xue[1], Ruhui Ma[1], Neil Robertson[2], Haibing Guan[1]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Queen's University Belfast, Belfast, UK
{hexufeng, zhangz-z-p, songt333, zhenguixue, ruhuima, hbguan}@sjtu.edu.cn, {Y.Hua, N.Robertson}@qub.ac.uk

ABSTRACT

With the rapid growth of video data, video summarization technique plays a key role in reducing people's efforts to explore the content of videos by generating concise but informative summaries. Though supervised video summarization approaches have been well studied and achieved state-of-the-art performance, unsupervised methods are still highly demanded due to the intrinsic difficulty of obtaining high-quality annotations. In this paper, we propose a novel yet simple unsupervised video summarization method with attentive conditional Generative Adversarial Networks (GANs). Firstly, we build our framework upon Generative Adversarial Networks in an unsupervised manner. Specifically, the generator produces high-level weighted frame features and predicts frame-level importance scores, while the discriminator tries to distinguish between weighted frame features and raw frame features. Furthermore, we utilize a conditional feature selector to guide GAN model to focus on more important temporal regions of the whole video frames. Secondly, we are the first to introduce the frame-level multi-head self-attention for video summarization, which learns long-range temporal dependencies along the whole video sequence and overcomes the local constraints of recurrent units, e.g., LSTMs. Extensive evaluations on two datasets, SumMe and TVSum, show that our proposed framework surpasses state-of-the-art unsupervised methods by a large margin, and even outperforms most of the supervised methods. Additionally, we also conduct the ablation study to unveil the influence of each component and parameter settings in our framework.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

video summarization, generative adversarial networks, video analysis, deep learning

*Corresponding author

1 INTRODUCTION

With the dramatic increase of the amount of video data, e.g., data from ubiquitous personal mobile phones or surveillance cameras, video summarization technique has attracted great attention recently. Video summarization aims to shorten the length of the input video while preserve enough information, which can still convey the whole storyline of the original video. It also can reduce tremendous human's efforts to explore video content and save huge amount of storage.

Most methods [7, 24, 34, 48, 50, 51] try to solve the video summarization problem in a supervised way, which mainly formulate video summarization as a sequence labeling problem and utilize the recurrent neural networks, such as LSTM, to capture temporal dependencies. Though these supervised methods can utilize the ground truth information in order to achieve the state-of-the-art performance, the intrinsic issues of these approaches are two-fold: 1) It is time-consuming to get ground truth summaries since annotators can only outline a video after watching the whole video; 2) Ground truth summaries created by different annotators may vary a lot from different perspectives and understandings of the video. Therefore, unsupervised video summarization approaches [25, 52] become popular and highly demanded recently.

In this paper, we propose a novel yet simple unsupervised video summarization method with attentive conditional Generative Adversarial Networks. The most closest work to our proposed framework is [25], which first applies GAN to unsupervised video summarization in a straight forward way. In [25], SUM-GAN is proposed with an encoder-decoder based GAN framework, which encodes the raw video frames to a subset, i.e., the summaries, and then reconstructs the video from summaries for discriminator to distinguish. However, there are several issues in SUM-GAN and the other similar GAN-based approaches: 1) The mapping from original video to summaries is information lossy, thus the reverse mapping is not always feasible [49]. For example, the objects which are in discard frames but not in summary frames can not be reconstructed solely based on summary frames; 2) SUM-GAN is complicated to be optimized and the variational autoencoder part requires pretraining; 3) Long-range temporal dependencies along the whole video are not captured to obtain global understanding of the video. In contrast, we utilize the adversarial learning [8] to learn the frame-level importance scores without any label. The generator of our proposed GAN model utilizes BiLSTM [9] to produce temporal representations of raw frame features (i.e., extracted by convolutional neural networks [44]) and predicts frame-level importance scores via these temporal representations. Then the generator produces the weighted frame features based on the temporal representations

weighted by the importance scores as fake inputs for the discriminator to distinguish from the real inputs (i.e., raw frame features). By alternatively training the generator and the discriminator, we can minimize the distance between these two adversarial frame features and force the generator to produce reasonable frame-level importance scores while make generated weighted frame features sufficiently similar to the raw frame features. The formulation of our GAN framework can solve the problems of information lossy mapping and the pre-training of VAE in SUM-GAN, since our framework maps raw frame features to weighted frame features to avoid sparse mapping and it does not contain the VAE part, thus it can be trained end-to-end. Additionally, we also utilize a conditional feature selector to provide conditional information for the GAN model to make it focus on more important subset frames of whole video frames motivated by [26, 43].

Moreover, as argued in [48], visually similar frames but with certain temporal distance should not be eliminated from the summaries. For example, in a football game, most penalty shootings are very similar in visual. But if these penalty shootings happen in different periods of the game, all of them should be kept in the final video summaries. Both SUM-GAN and several previous methods merely rely on LSTM to learn temporal dependencies among frames, which fails to learn long-range temporal dependencies of the whole video sequence as discussed in [40]. We are the first to introduce a multi-head self-attention module [39] to capture such long-range temporal dependencies between frames and ensure that the encoded temporal representations can contain enough temporal information of the whole video.

The contributions of this paper can be summarized as follows:

- We propose a novel yet simple unsupervised GAN framework for video summarization. The proposed GAN model can be easily extended to the conditional GAN model by utilizing conditional information provided by a conditional feature selector to focus on more important frames of the input video.
- To the best of our knowledge, we are the first to integrate the multi-head self-attention mechanism into a video summarization framework in order to capture long-range temporal dependencies.
- We surpass the state-of-the-art unsupervised methods by a large margin on two widely used datasets, i.e., SumMe and TVSum. Furthermore, our supervised variation also achieves competitive results comparing with recent approaches.

2 RELATED WORK

2.1 Video Summarization

The target of video summarization is to produce a shorter version of the original video which can still convey enough information as the original one and capture important events of the video. There are various ways to formulate the problem, such as video synopsis [33], time-lapses [15, 31], montage [16, 36] and storyboards [11, 12, 22, 23, 47]. Our work is closest to storyboards, which usually select a few video frames to summarize important events in the whole video.

Recently, deep learning based methods have achieved great improvement in video summarization. Zhang *et al.* [48] formulated the video summarization as a structured prediction problem and used BiLSTM to model long-range temporal dependencies among video frames. They further enhanced the LSTM-based model with a Determinantal Point Process (DPP) to increase the diversity of summaries. To overcome the deficiency of the DPP which treats video frames as randomly permutable items, SeqDPP [7] proposed a probabilistic model for the selection of diverse sequential subsets, which can heed the inherent sequential structures in videos. DySeqDPP [24] enabled SeqDPP to learn to automatically infer the local diversity degree from the input video, by introducing a latent variable. Zhang *et al.* [49] proposed a semantic loss between an original video and the video subset to evaluate the quality of video summarization. Zhou *et al.* [52] applied reinforcement learning to video summarization by designing a new reward function, which jointly accounts for diversity and representativeness of the generated video summaries. H-RNN [50] utilized two layer RNNs to learn temporal dependency among video frames, where the first layer RNN encodes short video sub-shots and the second layer predicts the scores of these sub-shots based on the first layer output. HSA-RNN [51] extended H-RNN with adaptively detecting shot boundary in the first layer. FCSN [34] formulated the video summarization as a sequence labeling problem and established a connection between semantic segmentation and video summarization. They also used fully convolutional sequence models instead of recurrent models. MAVS [6] proposed a memory augmented network to facilitate global understanding of whole video frames and summarize the video. Vasudevan *et al.* [38] solved query-relevant video summarization by calculating query-relevance and summarizing videos using query-relevance, diversity and representativeness for optimization.

2.2 Generative Adversarial Networks

A Generative Adversarial Network (GAN) [8] usually consists of two components, a generator and a discriminator. Following the adversarial training manner, the generator tries to generate fake data to confuse the discriminator, while the discriminator is designed to distinguish between the real data and the fake one. Due to the merit of GAN, it has achieved great success in various image and video processing tasks, such as image super-resolution [21], text-to-image synthesis [43, 46], image-to-image translation [13] and video prediction [28]. Mirza *et al.* [29] extended GAN to a conditional model which can control the data generated by GAN via concatenating conditional information with inputs of both generator and discriminator. Mahasseni *et al.* [25] first utilized GAN to solve the video summarization problem. They proposed a model based on VAE [19] and GAN for unsupervised video summarization by selecting a subset of key frames. Unlike previous unsupervised approaches, we propose a self-attention based conditional GAN framework to directly minimize the distance between generated weighted frame features and the raw frame features.

2.3 Attention models

The attention mechanism is powerful for capturing global dependencies. It has been widely used in machine translation [2], image generation [10] and image caption generation [42]. Self-attention modules [39] compute the response at a position by attending all positions in the whole sequence. Vaswani *et al.* [39] achieved state-of-the-art results in machine translation using the self-attention module. Wang *et al.* [40] bridged the self-attention with more general

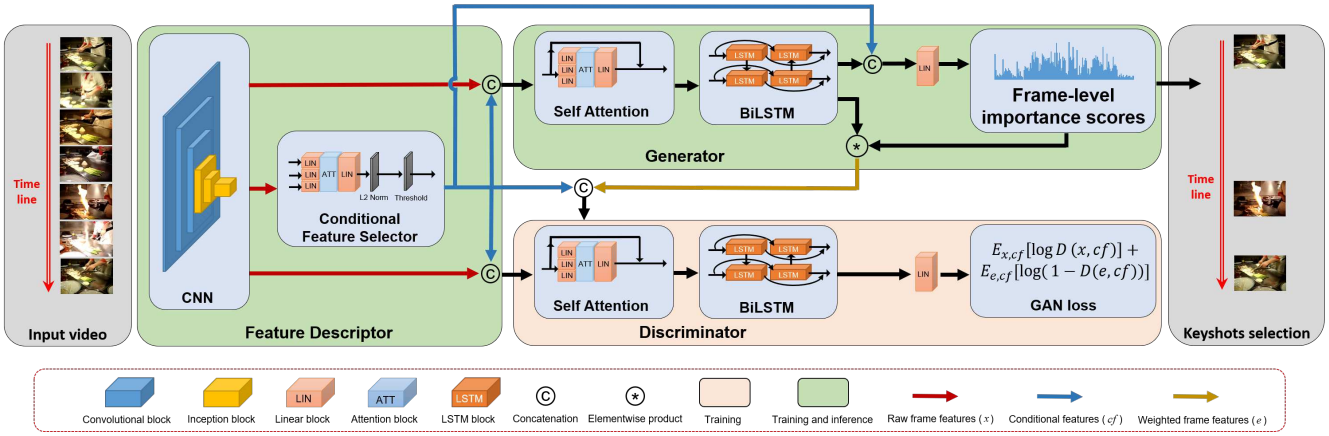


Figure 1: An overview of our proposed framework. When receiving a video, a CNN extracts frame feature vectors from the video for generator to predict frame importance scores which are used to generate summaries of the video via keyshot selection. The discriminator here tries to distinguish the output of generator and the raw frame features. The whole network is trained via adversarial loss without any labels. The multi-head self-attention module is described in section 3.2. The conditional feature selector is described in section 3.3. *Best viewed in color.*

classes of non-local operations, which modeled spatial-temporal dependencies. Zhang *et al.* [45] utilized the self-attention mechanism to model long-range dependencies in an image and showed great success in image generation. Wu *et al.* [41] proposed a 3-D part alignment model to learn local features via the attention mechanism for video-based person reidentification.

There are several papers utilizing attention in the video summarization area. Previous papers [5, 27] adopted attention, which is calculated from low-level features (e.g., motion and saliency map), as a cue to summarize the video. But these methods only rely on low-level features and they do not take the temporal dependencies among frames into consideration. The recent deep learning based method [14] extended the attention mechanism in [2] to video summarization. However, the attention is only used to re-weight the frame features so that the model can focus on a subset of frames. In contrast, the motivation of our multi-head self-attention module is to capture long-range temporal dependencies, since such information is important for generating good summaries, e.g., some repeated actions in the video sequences. To our best knowledge, we are the first to utilize multi-head self-attention in video summarization, which captures the long-range temporal dependencies and facilitates our framework with global information of the video.

3 METHOD

Our proposed unsupervised framework consists of a generator and a discriminator, as illustrated in Figure 1. The generator tries to predict the frame-level importance score for each frame and produces weighted frame features (i.e., e in Figure 1) based on temporal representations of raw frame features weighted by importance scores. Then, raw frame features (i.e., x in Figure 1) and weighted frame features are treated as real and fake inputs for the discriminator to distinguish. A multi-head self-attention module is introduced in this work to capture long-range temporal dependencies throughout the whole video sequence as a complementary guidance to the BiLSTM [9]. In order to make the GAN model focus on more important temporal regions of the whole video frames, we use selected features (i.e., cf in Figure 1) produced by the conditional feature selector as conditional input for the GAN model. We only use the feature descriptor and generator in inference, as shown by the light green box in Figure 1.

3.1 The GAN Framework and Loss

Our motivation is to train a network to identify the importance of different frames in a video, without providing any label. Thus, we utilize the adversarial learning method to provide guidance, i.e., a loss function, to help guiding the network to learn such importance score. The frame-level importance score reflects the likelihood that a frame should be included in the summary. Finally the summarization video can be generated based on the importance score. Additionally, we can extend the classical GAN model to the conditional GAN model [29] by utilizing selected features produced by the conditional feature selector depicted in section 3.3.

We first use a convolutional neural network (CNN) to extract visual features, i.e., raw frame features, from input video frames. Then, we feed the raw frame features to our GAN framework. We use two BiLSTMs [9] combined with two multi-head self-attention modules as the generator and the discriminator in our model. Raw frame features are fed into the generator, and the generator produces weighted frame features as the fake inputs for the discriminator to distinguish from real inputs (i.e., raw frame features). To provide enough conditional information for guiding the GAN training, we concatenate conditional features cf with the input and BiLSTM output of the generator. We also concatenate cf with both the real and fake inputs of the discriminator.

The generator and the discriminator are trained alternatively. The raw frame features and the weighted frame features from the generator are fed into the discriminator, and the loss is calculated to update the parameters of the discriminator. The same weighted frame features are fed into the discriminator to calculate the loss

and update the parameters of the generator. This process is repeated until the end of the training.

Generator. The generator G consists of one multi-head self-attention module and one BiLSTM model. The outputs of G are two branches. One branch is used to generate the weighted frame features, which have the same shape as the raw frame features. The other branch is connected with two fully connected layers and a sigmoid activation function (i.e., linear block in Figure 1) to obtain predicted frame-level importance scores.

More specifically, given raw frame features $x = \{x_t : t = 1, ..., T\} \in \mathbb{R}^{T\times d}$ and conditional features $cf = \{cf_t : t = 1, ..., T\} \in \mathbb{R}^{T\times d}$, with T being the number of input frames and d being the dimension of each frame feature, the concatenation of x and cf is fed into the generator. Temporal representations produced by the generator have the same shape as the raw frame features, denoted by $f = \{f_t : t = 1, ..., T\} \in \mathbb{R}^{T\times d}$. We can obtain the normalized predicted importance scores $s = \{s_t : s_t \in (0, 1), t = 1, ..., T\} \in \mathbb{R}^T$ by feeding temporal representations f concatenated with cf to fully connected layers. The weighted frame features produced by the generator are denoted as $e = \{e_t : e_t = s_t \times f_t, t = 1, ..., T\} \in \mathbb{R}^{T\times d}$.

Discriminator. The discriminator D consists of one multi-head self-attention module, one BiLSTM model followed by two fully connected layers and a sigmoid activation function (i.e., linear block in Figure 1) to produce discriminator scores, i.e., labelling 1 for real inputs and 0 for fake inputs. Similar to the classical conditional GAN framework [29], we view the discriminator as a classifier which aims to distinguish raw frame features x and weighted frame features e conditioned on conditional features cf. The function of discriminator here is to estimate the similarity between two features and force the generator to generate weighted frame features that are sufficiently similar to the raw frame features. Once the training completes, predicted the frame-level importance scores reflect the importance of frames in a video.

Adversarial Loss. The proposed framework adopts the original GAN loss [8] (i.e., sigmoid cross entropy loss) and applies spectral normalization [30] on fully connected layers of both the generator and the discriminator [20, 45]. Given raw frame features x, conditional features cf and weighted frame features e, the min-max adversarial learning loss is defined as

$$\begin{aligned}\min_{\mathrm{G}}\max_{\mathrm{D}} L(G, D) &= \mathbb{E}_{x,cf}[\log D(x, cf)] + \mathbb{E}_{x,cf}[\log(1 - D(G(x), cf))] \\ &= \mathbb{E}_{x,cf}[\log D(x, cf)] + \mathbb{E}_{e,cf}[\log(1 - D(e, cf))].\end{aligned} \tag{1}$$

During the training process, the generator G is trained by maximizing $\log D(e, cf)$ instead of minimizing $\log(1 - D(e, cf))$ to avoid saturating gradients for the generator. The discriminator D and the generator G are alternatively updated. While updating D, the parameters of G are reused. The objective function of discriminator D is defined as

$$\min_{\mathrm{D}} L(D) = -\Big(\mathbb{E}_{x,cf}[\log D(x, cf)] + \mathbb{E}_{x,cf}[\log(1 - D(G(x), cf))]\Big). \tag{2}$$

When updating the generator G, the objective function of G is defined as

$$\min_{\mathrm{G}} L(G) = -\mathbb{E}_{x,cf}[\log D(G(x), cf)]. \tag{3}$$

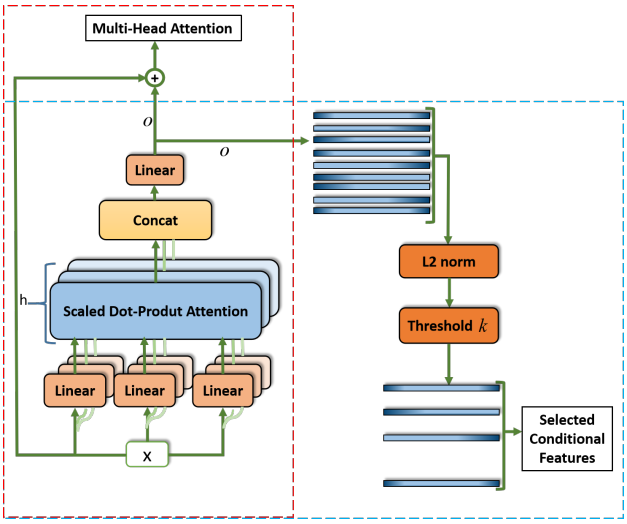


Figure 2: Illustration of Multi-Head Self-Attention module (dashed red box) and Conditional Feature Selector (dashed blue box). x is the frame features, h denotes the number of heads, k is the threshold parameter which controls the number of selected feature vectors. Each head performs dot-product attention and their outputs are concatenated and then linearly projected. For multi-head self-attention module, the projected features o is added with x to get the multi-head attention. For conditional feature selector, feature vectors with top-k $L2$-norm in o are selected as conditional features. *Best viewed in color.*

3.2 Multi-Head Self-Attention Module

To generate diverse and compact summary of a video needs global understanding of the whole video, thus information such as long-range temporal dependencies among frames is critical for video summarization. One of the challenges in video summarization is how to learn long-range temporal dependencies along the whole video sequence. Most video summary models [25, 48, 52] merely rely on recurrent units like LSTMs to capture such dependencies of a video. However, recurrent units are only able to process one local neighbourhood at a time [40]. Long-range temporal dependencies are obtained by repeatedly applying recurrent operations. Thus, recurrent units may fail to model long-range temporal dependencies of the whole video sequence. A multi-head self-attention module [39] computes the response at a position by attending all positions in the whole sequence, which provides information about how frames are related in a video. In this section, we introduce our frame-level multi-head self-attention module for video summarization, motivated by [39, 45].

The proposed multi-head self-attention module is shown in Figure 2. Each self-attention head performs the same operation. The reason for the use of multi-head here is that each head may attend different positions of the whole sequence, which provides more diverse temporal information. One thing should be noticed is that if the GAN model utilizes conditional features cf, then the input features is the concatenation of x and cf with the shape $T\times 2d$. However, the computation process is the same, thus we use raw frame

features x as input to describe the module for simplicity. Given raw frame features x, we feed them to each head self-attention module. Raw frame features x are first transformed to two feature spaces f, g via fully connected layers to calculate the attention map. The attention map $a \in \mathbb{R}^{T\times T}$ is obtained via

$$a = \text{softmax}\left(\frac{f(x)g(x)^T}{\sqrt{d'}}\right), \tag{4}$$

where $f(x) = xW_f$, $W_f \in \mathbb{R}^{d\times d'}$, and $g(x) = xW_g$, $W_g \in \mathbb{R}^{d\times d'}$. $d' = d/h$, d is the dimension of raw frame features x and h is the number of heads. The softmax operation is performed along each row.

To prevent dot products from growing too large, we scale dot products using the same way as [39]. Thus a_{ij} in the attention map a reflects how frame i and frame j are related in input frame features. The whole attention map can get the global information of the whole input sequence. The output feature matrix of each head is obtained by

$$head_i = a * k(x), i = 1, ..., h, \tag{5}$$

where $k(x) = xW_k$, $W_k \in \mathbb{R}^{d\times d'}$, $head_i \in \mathbb{R}^{T\times d'}$. The concatenation of each head is then linearly projected to get o, where $o = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W_h$, $W_h \in \mathbb{R}^{d\times d}$ and, $o \in \mathbb{R}^{T\times d}$. The final output of the multi-head self-attention module is obtained by

$$\text{y} = o + x. \tag{6}$$

3.3 Conditional Feature Selector

Since there always exist more important temporal regions of the whole video frames, it is beneficial for generating higher quality summary if the model can focus on such regions first and then generate more diverse summary. Motivated by utilizing attention as conditional information to guide GAN training [43], we select part of features from o via $L2$-norm of these feature vectors [26] as conditional information of video to guide the GAN network focusing on more important subset frames of the whole video frames. The conditional feature selector only operates on raw frame features x to produce conditional features cf. Since this module has an overlap with multi-head self-attention module, some expressions in section 3.2 will be reused for simplicity.

More specifically, each head output h_i is the re-weighted features obtained via the self-attention mechanism on each head, and the concatenation of h_i after linear projection is the projected features o. Since o has the shape as the raw frame features x and reveals different importances of T frames obtained by the self-attention mechanism, we select the k frames in o by the $L2$-norm along each row, denoted by $l \in \mathbb{R}^T$ in which

$$l_t = \|o_t\|_2 \in \mathbb{R}, t = 1, ..., T, \tag{7}$$

with $\|\cdot\|_2$ denoting $L2$-norm. We select the k frame features from o which have the top-k $L2$-norm in l. The k selected indices $\hat{l} = [l_1, ..., l_k]$ are used to select features from o, the final selected frame features is $c \in \mathbb{R}^{k\times d}$. However, in order to make the whole model differential, we use the k indices to produce a boolean mask m which has the same shape as x with all ones in selected rows. Thus $c = m * o, c \in \mathbb{R}^{T\times d}$. Once we get the selected frame features c, c is fed into the GAN framework as conditional information to make the network pay more attention to these selected frames. Typically, the threshold parameter k is determined by the final summary length of the original video. In this paper, we set $k = 0.15 \times T$, following the summary length is 15% of the original video length used in previous papers by convention.

3.4 Supervised extension

We extend our unsupervised model to a supervised version by introducing a least square loss between predicted frame-level importance scores and ground-truth frame-level importance scores. It acts as the regularization when updating the generator, which is defined as follows

$$L2_{reg} = \frac{1}{T}\sum_t \|s_t - g_t\|^2, \tag{8}$$

where $s = \{s_t : t = 1, ..., T\} \in \mathbb{R}^T$ are predicted frame-level importance scores and $g = \{g_t : t = 1, ..., T\} \in \mathbb{R}^T$ are ground-truth frame-level importance scores.

4 EXPERIMENTAL RESULTS

4.1 Evaluation dataset

We evaluate the performance of our proposed framework on two commonly used benchmark datasets, SumMe [12] and TVSum [35]. The SumMe dataset contains 25 diverse videos which include multiple events such as sports, holiday, and cooking, etc. These videos are mostly 1.5 minutes to 6.5 minutes in length. The TVSum dataset contains 50 videos downloaded from YouTube in 10 categories. The video lengths vary from 1 to 5 minutes. Both datasets provide frame-level importance scores. In addition, we also use the YouTube [3] dataset and the Open Video Project (OVP) dataset [48] to augment the training data. The YouTube dataset contains 39 videos excluding cartoon videos and the OVP dataset contains 50 videos in different categories, e.g., documentary.

4.2 Evaluation metrics and settings

By convention, we use key-shot-based F-score [48] as the metric to assess the similarity between automatically generated summaries and ground truth summaries. Denote A as the generated key-shot summary, which contains less than 15% duration of the original video. Denote B as the user-annotated key-shot summary. We first compute the precision and recall for A against B for evaluation according to the temporal overlap between two summaries, as follows

$$\text{Precision (P)} = \frac{\text{overlapped duration of A and B}}{\text{duration of A}} \tag{9}$$

$$\text{Recall (R)} = \frac{\text{overlapped duration of A and B}}{\text{duration of B}}. \tag{10}$$

Then, the final harmonic mean F-score can be obtained by

$$\text{F} = 2\text{P} \times \text{R}/(\text{P} + \text{R}) \times 100\%. \tag{11}$$

Following [35, 48, 52], we convert frame-level importance scores to key-shot summaries for evaluation. In order to generate key-shot summaries from frame-level importance scores, we first use KTS [32] to temporally segment the video into disjoint intervals. Then, we compute the average score of each interval, and assign each frame in the interval with this average score. Finally, we rank

frames by their average scores and use knapsack algorithm to select frames so that the total length is under a threshold, usually 15% of the video length. In this way, we can get the key-shot summaries for the performance evaluation.

For a fair comparison and clear analysis, we highlight the differences of test methods and evaluation settings. For test methods, according to previous work, some randomly split the dataset to 80% videos for training and 20% videos for testing, and calculate the average F-score over multiple trials. This test method may cause overlap in testing videos of different trials. We name such test method according to the number of random splits, e.g., 5 Random, 10 Random, and Multiple Random. Meanwhile, we prefer the 5-fold cross validation (5FCV) test method. With 5FCV, the model can be evaluated on all videos of the dataset, which makes the results more reliable and reproducible for future comparison. The standard 5FCV is used as our default test method in the following discussion. For evaluation settings, similar to [25, 48, 52], we evaluate and compare our method with four different evaluation settings:

(1) **Canonical.** We report the average F-score with 5FCV on each dataset individually.
(2) **Augmented.** We augment each fold of 5FCV with the other three datasets during training.
(3) **Transfer.** This challenging task is introduced by [48] to test the transfer ability of the model, which is relevant for practical applications. We train our model on three datasets and test our model on the remaining one, e.g., train on YouTube, OVP and TVSum, then test on SumMe.
(4) **One-to-one Transfer.** Adopting the similar idea of "Transfer" evaluation setting, we denote the strategy of training the model only on one dataset, and test on the other datasets as One-to-one Transfer evaluation setting.

4.3 Implementation details

For fair comparison with other methods, the feature descriptor of each frame is extracted via the output of *pool*5 layer of GoogLeNet [37] model (1024-dimensions), pre-trained on ImageNet [4]. For the generator, we use one BiLSTM with 1024 hidden units followed by a 256-dimensional fully connected layer and a 1-dimensional sigmoid layer to get the predicted scores. For the discriminator, we use one BiLSTM with 512 hidden units followed by a 256-dimensional fully connected layer and a 1-dimensional sigmoid layer to get the discriminator scores. For both the generator and the discriminator, we apply spectral normalization [30] on fully connected layers. We implement our model using Tensorflow [1] and train the model on one GTX 1080Ti graphics card. We set the learning rate as 0.00004 for the discriminator and 0.00001 for the generator. We use Adam optimizer [18] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for training.

4.4 Ablation study

In this section, we do various ablation analysis to evaluate the impact of different components and parameter settings of our unsupervised model on canonical evaluation setting.

The impact of different components of the framework. We first conduct ablation studies to unveil how different components influence the performance of the whole framework shown in Table 1. The different model settings are as followed:

Model	SumMe	TVSum
ACGAN$_{base}$	43.7	57.6
ACGAN$_{w/o-SA}$	44.9	56.0
ACGAN$_{w/o-CF}$	44.2	57.8
ACGAN	**46.0**	**58.5**

Table 1: Performance (F-score, %) with our framework with different components on SumMe and TVSum.

- **ACGAN**: The proposed **a**ttentive **c**onditional GAN model.
- **ACGAN**$_{w/o-SA}$: Drop the multi-head **s**elf-**a**ttention module (SA) in ACGAN.
- **ACGAN**$_{w/o-CF}$: Drop **c**onditional **f**eature selector (CF) in ACGAN.
- **ACGAN**$_{w/o-\{SA,CF\}}$: This is our GAN baseline model without integrating multi-head self-attention module and conditional feature selector. We also denote this model as **ACGAN**$_{base}$ in the remaining paper for simplicity.

From Table 1, comparing ACGAN$_{w/o-SA}$, ACGAN$_{w/o-CF}$ and ACGAN, we observe that both the multi-head self-attention module and the conditional feature selector are important for our ACGAN model. The multi-head self-attention module facilitates BiLSTM by providing long-range temporal dependencies among frames. The conditional feature selector can help the ACGAN model by guiding it focusing on more important subset of all frames. For ACGAN$_{base}$, we notice that the multi-head self-attention module improves the performance on both datasets. However, ACGAN$_{w/o-SA}$ underperforms ACGAN$_{base}$ on TVSum dataset. The reason is that the conditional feature selector makes the model focus on subset of all frames which may decrease summary diversity. But this also reflects that we can control the GAN baseline ACGAN$_{base}$ with conditional information. When utilizing both modules, the multi-head self-attention module can force the model to generate more diverse summary and solve the problem introduced by the conditional feature selector. We use ACGAN$_{base}$ and ACGAN for comparisons with state-of-the-art methods.

The impact of the input video clip length versus the number of heads. We analyze how the performance of ACGAN is related to the length of the input video clip versus the number of heads in multi-head self-attention module and the conditional feature selector. For input video clip length, we choose T=160, T=320, T=640 and variable length (i.e., the length of original videos) for comparison under canonical evaluation setting. For the number of heads, we choose 1 head, 4 heads and 8 heads for comparison under canonical evaluation setting. From Table 2 and Table 3, we observe that for fixed input length, our model with different numbers of heads obtains best performance on two datasets when T=320. For variable input length, we notice that there exists performance degradation on both SumMe and TVSum datasets. The reason is that the model can get more complete information about the entire video when the input length is variable, however, this causes the size of attention map to change during the training process and the changing attention map cannot facilitate the whole model as well as the fixed input length. Therefore, in our model, we randomly select continuous 320 frames (i.e., T=320) in input video sequence for training; While for testing, we feed the whole video sequence to the network as other methods do for a fair comparison.

Number of heads	T=160	T=320	T=640	Variable length
1 head	43.1	44.4	43.8	43.6
4 heads	44.7	**46.2**	44.5	43.9
8 heads	44.4	46.0	45.8	44.2

Table 2: Performance (F-score, %) of ACGAN with different input video clip lengths versus different numbers of heads on SumMe dataset

Number of heads	T=160	T=320	T=640	Variable length
1 head	55.4	55.9	55.7	55.4
4 heads	57.4	57.7	57.1	57.3
8 heads	58.0	**58.5**	57.4	57.7

Table 3: Performance (F-score, %) of ACGAN with different input video clip lengths versus different numbers of heads on TVSum dataset

Model	SumMe	TVSum	Test methods
Video-MMR[23]	26.6	-	-
Vsumm[3]	33.7	-	-
Web image[17]	-	36.0	-
Co-archetypal[35]	-	50.0	-
SUM-GAN$_{dpp}$[25]	39.1	51.7	5 Random
DR-DSN[52]	41.4	57.6	5FCV
SUM-FCN$_{unsup}$[34]	41.5	52.7	Multiple Random
ACGAN$_{base}$	43.7	57.6	5FCV
ACGAN	**46.0**	**58.5**	5FCV

Table 4: Performance (F-score, %) of our method and other unsupervised approaches on SumMe and TVSum. Our model performs the best on both datasets

As illustrated in Table 2 and Table 3, we notice that the number of heads in the multi-head self-attention module and the conditional feature selector is critical for the overall performance under different input video clips lengths, comparing the performance between 1 head and 4/8 heads. The main reason is that more heads in the multi-head structure allow the model to jointly attend to information from different representation sub-spaces of the whole video frames. In our model, we set all multi-head self-attention modules with 8 heads.

4.5 Comparison with state-of-the-art

Comparison with unsupervised approaches. Table 4 shows the results of our model against other unsupervised methods. Our model outperforms the state-of-the-art models on both datasets, specifically, ACGAN is 10.8% better than SUM-FCN$_{unsup}$ on SumMe, and 1.7% better than DR-DSN on TVSum. In addition, ACGAN$_{base}$ also outperforms current unsupervised methods on both datasets. This observation shows that our simple ACGAN$_{base}$ without any additional modules and regularizers is more powerful than previous GAN model SUM-GAN$_{dpp}$ [25].

Comparison with supervised approaches. Table 5 shows the results of our supervised model, denoted as ACGAN$_{sup}$, against other supervised approaches. For both SumMe and TVSum datasets, our supervised model outperforms most supervised methods, but is slightly lower than SUM-FCN and SUM-DeepLab on SumMe, and MAVS on TVSum. It is worth noting that these methods mainly achieve great performance improvement only on one dataset but perform much worse on the other dataset while our method can provide reasonable performance improvement on both datasets. Furthermore, we can also observe that both of our unsupervised model ACGAN from Table 4 and supervised GAN baseline ACGAN$_{base-sup}$ outperform most of the supervised methods, as shown in Table 5.

Model	SumMe	TVSum	Test methods
Interestingness[12]	39.4	-	-
Submodularity[11]	39.7	-	-
Summary transfer[47]	40.9	-	Multiple Random
Bi-LSTM[48]	37.6	54.2	-
DPP-LSTM[48]	38.6	54.7	-
SUM-GAN$_{sup}$[25]	41.7	56.3	5 Random
DR-DSN$_{sup}$[52]	42.1	58.1	5FCV
MAVS[6]	43.1	**67.5**	5FCV
DySeqDPP[24]	44.3	58.4	10 Random
SUM-FCN[34]	47.5	56.8	Multiple Random
SUM-DeepLab[34]	**48.8**	58.4	Multiple Random
ACGAN$_{base-sup}$	44.8	58.3	5FCV
ACGAN$_{sup}$	47.2	59.4	5FCV

Table 5: Performance (F-score, %) of our supervised method and other supervised approaches on SumMe and TVSum.

Transfer and augmented results. We also compare the performance in augmented and transfer evaluation settings of our model ACGAN against other unsupervised approaches, as shown in Table 6. We observe that our unsupervised method achieves the best performance under transfer evaluation setting on both datasets. For augmented evaluation setting, our model also achieves the highest F-score on SumMe but underperforms SUM-GAN$_{dpp}$ [25] slightly on TVSum. We conclude two reasons: 1) SUM-GAN$_{dpp}$ uses the 5 Random test method, i.e., randomly spliting the dataset and reporting the average F-score, while we use 5FCV, which generally results in lower performance, but is more reliable and reproducible. 2) SUM-GAN$_{dpp}$ uses heavy regularization, e.g., dpp and repelling, to provide additional information with more data.

In order to further evaluate the transfer ability of ACGAN, we perform experiments on One-to-one Transfer evaluation setting, as shown in Table 7. We find our model achieves appealing performance in various dataset combinations. We notice that as the training dataset gets larger, the performance on the target dataset also improves. For example, when the model is trained on YouTube (39 videos), or OVP (50 videos), or TVSum (50 videos), and tested on SumMe, the results are 42.6%, 43.9% and 44.0%, respectively. Notice that OVP and TVSum contain more videos than YouTube (50 for OVP or TVSum, versus 39 for YouTube). Such appealing property proves that our model is useful for practical applications. For example, we may utilize lots of unlabeled data and obtain reasonable performance in fully unsupervised manner on the target dataset which the model did not see before.

4.6 Qualitative results

Video summaries. We visualize the ground truth importance scores and selected frames of our model in Figure 3 to give an intuitive understanding of our model. All four variants of our model

(a) ACGAN$_{sup}$, F-score: 70.8

(b) ACGAN, F-score: 68.8

(c) ACGAN$_{w/o-CF}$, F-score: 68.7

(d) ACGAN$_{base}$, F-score: 65.5

Figure 3: Video summaries generated by different variants of our approach of video 18 in TVSum. The grey bars show the ground truth importance scores, the colored areas are the selected frames by different models. The F-scores (%) of each model are shown below each image. *Best viewed in color.*

Model	Test methods	SumMe		TVSum	
		A	T	A	T
SUM-GAN$_{dpp}$[25]	5 Random	43.4	-	**59.5**	-
DR-DSN[52]	5FCV	42.8	42.4	58.4	**57.8**
ACGAN	5FCV	**47.0**	**44.5**	58.9	**57.8**

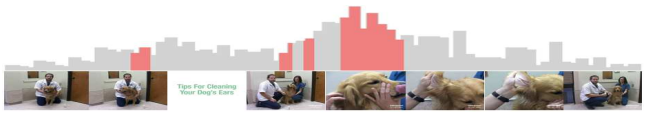
Table 6: Performance (F-score, %) of our unsupervised method and other unsupervised approaches under Augmented (A) and Transfer (T) evaluation settings on SumMe and TVSum.

Test dataset	Train dataset			
	YouTube	OVP	SumMe	TVSum
SumMe	42.6	43.9	-	44.0
TVSum	56.3	57.3	56.6	-

Table 7: Performance (F-score, %) of our unsupervised model using different combinations of training and testing datasets, on One-to-one Transfer evaluation setting.

generate high quality summaries of this video despite small variations. The generated summaries are diverse and they capture almost all peak regions of the ground truth scores. The supervised model (see Figure 3(a)) benefits from the labels, and generates higher quality summaries as well as capturing peak regions of the ground truth scores better than other unsupervised variations, especially at the end of the video.

Selected indices $\hat{l}$ of the conditional feature selector. We also use selected indices $\hat{l}$ as the frame-level importance scores, i.e., label selected indices with 1 and others with 0, to generate the summary. An illustration can be found in Figure 4. From Figure 4(a), we find that we can still generate reasonable summary even directly using the selected indices $\hat{l}$. In contrast, the generated summary in Figure 4(b) is more diverse than that in Figure 4(a). At first, the selected conditional features make the model focus on peak regions of the ground truth scores. In later frames, the multi-head self-attention mechanism captures long-range temporal dependencies along the whole video frames and forces the model to generate more diverse summary, reflecting the storyline of the original video more completely.



(a) Selected indices as frame-level importance scores, F-score: 63.4

(b) Original frame-level importance scores, F-score: 71.9

Figure 4: Video summaries generated by the selected indices in the multi-head self-attention module (see (a)) and the original frame-level importance scores (see (b)) of video 15 in TVSum dataset. Grey bars show the ground truth importance scores, while the colored areas are the selected frames. The F-scores (%) are shown below each image. *Best viewed in color.*

5 CONCLUSION

This paper presents an unsupervised attentive conditional generative adversarial network for video summarization. The generator predicts frame-level importance scores and produces weighted frame features. The discriminator tries to distinguish the weighted frame features and the raw frame features. The model is trained in an adversarial manner to minimize the distance between the weighted frame features and the raw frame features. A conditional feature selector is proposed to provide conditional information for the GAN model to make it focus on more important temporal regions of the whole video frames. Moreover, the multi-head self-attention module is incorporated to both generator and discriminator due to its ability for capturing long-range temporal dependencies. Experimental results show our model outperforms the other state-of-the-art unsupervised approaches on two benchmark datasets by a large margin. The supervised variation of our model also achieves competitive results comparing with recent approaches.

ACKNOWLEDGMENT

This work was supported in part by National NSF of China (NO. 61525204, 61732010, 61872234) and Shanghai Key Laboratory of Scalable Computing and Systems.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *OSDI*.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

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

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

[5] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik. 2013. Efficient visual attention based framework for extracting key frames from videos. *Signal Processing: Image Communication* 28, 1 (2013), 34–44.

[6] Litong Feng, Ziyin Li, Zhanghui Kuang, and Wei Zhang. 2018. Extractive Video Summarizer with Memory Augmented Neural Networks. In *MM*.

[7] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse Sequential Subset Selection for Supervised Video Summarization. In *NIPS*.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[9] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (2005), 602–610.

[10] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A Recurrent Neural Network For Image Generation. In *ICML*.

[11] Michael Gygli, Helmut Grabner, and Luc J. Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *CVPR*.

[12] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc J. Van Gool. 2014. Creating Summaries from User Videos. In *ECCV*.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.

[14] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. 2019. Video Summarization with Attention-Based Encoder-Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology* (2019).

[15] Neel Joshi, Wolf Kienzle, Mike Toelle, Matt Uyttendaele, and Michael F. Cohen. 2015. Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics* 34, 4 (2015), 63:1–63:9.

[16] Hong-Wen Kang, Yasuyuki Matsushita, Xiaoou Tang, and Xue-Quan Chen. 2006. Space-Time Video Montage. In *CVPR*.

[17] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. 2013. Large-Scale Video Summarization Using Web-Image Priors. In *CVPR*.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. 2018. The gan landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720* (2018).

[21] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR*.

[22] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. 2012. Discovering important people and objects for egocentric video summarization. In *CVPR*.

[23] Yingbo Li and Bernard Mérialdo. 2010. Multi-video summarization based on Video-MMR. In *WIAMIS*.

[24] Yandong Li, Liqiang Wang, Tianbao Yang, and Boqing Gong. 2018. How Local Is the Local Diversity? Reinforcing Sequential Determinantal Point Processes with Dynamic Ground Sets for Supervised Video Summarization. In *ECCV*.

[25] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised Video Summarization with Adversarial LSTM Networks. In *CVPR*.

[26] Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. 2018. Learning Visual Question Answering by Bootstrapping Hard Attention. In *ECCV*.

[27] Sophie Marat, Mickael Guironnet, and Denis Pellerin. 2007. Video summarization using a visual attention model. In *EUSIPCO*.

[28] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).

[29] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014).

[30] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[31] Yair Poleg, Tavi Halperin, Chetan Arora, and Shmuel Peleg. 2015. EgoSampling: Fast-forward and stereo for egocentric videos. In *CVPR*.

[32] Danila Potapov, Matthijs Douze, Zaïd Harchaoui, and Cordelia Schmid. 2014. Category-Specific Video Summarization. In *ECCV*.

[33] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. 2008. Nonchronological Video Synopsis and Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 11 (2008), 1971–1984.

[34] Mrigank Rochan, Linwei Ye, and Yang Wang. 2018. Video Summarization Using Fully Convolutional Sequence Networks. In *ECCV*.

[35] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. TVSum: Summarizing web videos using titles. In *CVPR*.

[36] Min Sun, Ali Farhadi, Ben Taskar, and Steven M. Seitz. 2014. Salient Montages from Unconstrained Videos. In *ECCV*.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*.

[38] Arun Balajee Vasudevan, Michael Gygli, Anna Volokitin, and Luc Van Gool. 2017. Query-adaptive Video Summarization via Quality-aware Relevance Estimation. In *MM*.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local Neural Networks. In *CVPR*.

[41] L. Wu, Y. Wang, L. Shao, and M. Wang. 2019. 3-D PersonVLAD: Learning Deep Global Representations for Video-Based Person Reidentification. *IEEE Transactions on Neural Networks and Learning Systems* (2019), 1–13.

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*.

[43] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks. In *CVPR*.

[44] LeCun Yann, Bottou Leon, Bengio Yoshua, and Haffner Patrick. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[45] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-Attention Generative Adversarial Networks. *arXiv preprint arXiv:1805.08318* (2018).

[46] Han Zhang, Tao Xu, and Hongsheng Li. 2017. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *ICCV*.

[47] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Summary Transfer: Exemplar-Based Subset Selection for Video Summarization. In *CVPR*.

[48] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video Summarization with Long Short-Term Memory. In *ECCV*.

[49] Ke Zhang, Kristen Grauman, and Fei Sha. 2018. Retrospective Encoders for Video Summarization. In *ECCV*.

[50] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2017. Hierarchical Recurrent Neural Network for Video Summarization. In *MM*.

[51] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2018. HSA-RNN: Hierarchical Structure-Adaptive RNN for Video Summarization. In *CVPR*.

[52] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep Reinforcement Learning for Unsupervised Video Summarization With Diversity-Representativeness Reward. In *AAAI*.