This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

Relation Attention for Temporal Action Localization

Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan

***Abstract*—Temporal action localization aims to accurately localize and recognize all possible action instances from an untrimmed video automatically. Most existing methods perform this task by first generating a set of proposals and then recognizing each independently. However, due to the complex structures and large content variations in action instances, recognizing them individually can be difficult. Fortunately, some proposals often share information regarding one specific action. Such information, which is ignored in existing methods, can be used to boost recognition performance. In this paper, we propose a novel mechanism, called relation attention, to exploit informative relations among proposals based on their appearance or optical flow features. Specifically, we propose a relation attention module to enhance representation power by capturing useful information from other proposals. This module does not change the dimensions of the original input and output and does not rely on any specific proposal generation methods or feature extraction backbone networks. Experimental results show that the proposed relation attention mechanism improves performance significantly on both Thumos14 and ActivityNet1.3 datasets compared to existing architectures. For example, relying on Structured Segment Networks (SSN), the proposed relation attention module helps to increase the mAP from 41.4 to 43.7 on the Thumos14 dataset and outperforms the state-of-the-art results.**

***Index Terms*—temporal action localization, relation attention.**

I. Introduction

In the past few years, deep learning has been widely used to analyze visual content [1]–[7], especially for action recognition [8], [9]. This task assumes that background instances are removed beforehand and mainly focus on classifying trimmed video clips. However, in practice, it is time-consuming and expensive to trim each video manually. Thus, it is desirable to localize the position of all possible action instances from untrimmed videos automatically and then recognize them. This task, known as temporal action localization, has various potential applications in content-based video searching [10], temporal sentence localization [11] and monitoring suspicious activities [12].

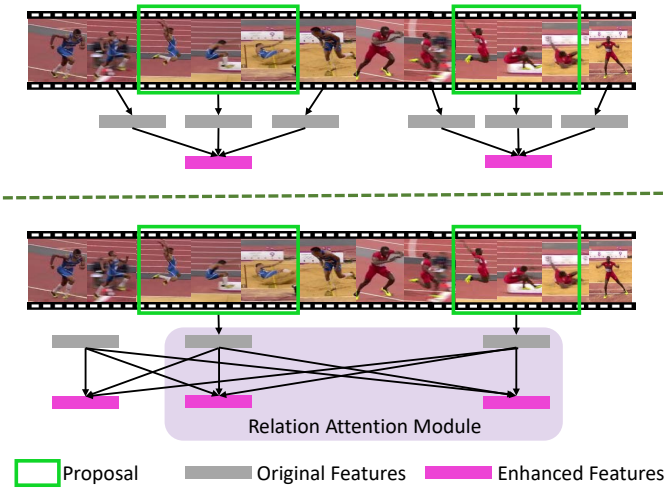

Fig. 1. Two different methods of capturing context information in a video. **Top**: For a proposal (green box), we extract features from the frames within and around it, and then concatenate them as augmented features. **Bottom**: We first extract features for all proposals and then put them into our relation attention module. Output enhanced features can be regarded as the weighted average of all input features based on the learned relations between proposals.

Temporal action localization is closely related to object detection in the image domain [13]–[16] because both try to find meaningful regions (2D bounding boxes in object detection and 1D time intervals in temporal action localization) and then recognize them. Thus, action localization can be viewed as a 1D counterpart of object detection. Inspired by the success of region-based paradigm established in R-CNN [17], most temporal action localization algorithms involve two stages: 1) generate proposals which are likely to contain actions; and 2) perform classification and boundary regression on each proposal individually. The second stage can be inspected as an action recognition task for each proposal, and the common approach is to extract proposal features first and then train a feature classifier with action label [18]. In this scene, the quality of proposal features is critical. To exploit high-quality features, several attempts, which extend the receptive fields when extracting proposal features, have been proposed [19]–[22]. These methods capture information from frames around the proposals and thus take more information into consideration, as shown on the top of Figure 1. However, this type of method still suffers from two main issues: 1) the range of sampling contextual information is restricted to a local area, and thus global contextual information is neglected; and 2) proposals are still recognized separately. The second issue always leads to a decrease in performance because

Manuscript received December 20, 2018; revised July 13, 2019 and November 12, 2019; accepted November 29, 2019. This work was partially supported by Guangdong Provincial Scientific and Technological Funds under Grants 2018B010107001, key project of NSFC (No. 61836003), National Natural Science Foundation of China (NSFC) 61602185, Program for Guangdong Introducing Innovative and Entrepreneurial Teams 2017ZT07X183, Tencent AI Lab Rhino-Bird Focused Research Program (No. JR201902), Fundamental Research Funds for the Central Universities D2191240, Open Project of State Key Laboratory of Subtropical Building Science for International Cooperation Research (No. 2019ZA01). *(Corresponding author: Mingkui Tan.)*

P. Chen and R. Zeng are with the School of Software Engineering, South China University of Technology, Guangzhou 510630, China (e-mail: phchencs@gmail.com; runhaozeng.cs@gmail.com).

M. Tan is with the School of Software Engineering, South China University of Technology, Guangzhou 510630, China; Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: mingkuitan@scut.edu.cn).

C. Gan is with MIT-IBM Watson AI Lab, Cambridge, MA 02142, USA (e-mail: ganchuang1990@gmail.com).

G. Shen and W. Huang are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: thusgy2012@gmail.com; hwenbing@126.com).

recognizing proposals individually can be difficult, due to the complex structures and large content variations in action instances. Fortunately, we observed that some proposals could share essential or complementary information regarding one specific action category. For example, both "ball serving" and "tennis smash" action instances can be classified as the "tennis swing" action class. Although these two kinds of instance have their distinct characteristics, they also share essential common features considering the "tennis swing" action. Considering several instances simultaneously allows the network to learn essential characteristics instead of overfitting the features of distinct action instance. Another example is that the video of "long jumping" usually consists of both background information (*e.g.,* sand pool) and actions (*e.g.,* jumping, running). Such information can be complementary and provides clues for temporal reasoning, which benefits the understanding of actions. This relative and helpful information may spread over the video. Therefore, the range of searching for context information should not be restricted locally. Inspired by [23], which leveraged relations between the detected semantic regions for scene parsing, we hope to exploit relations between action instances for temporal action localization task.

To model a relation between proposals, we propose a relation attention module for temporal action localization. As illustrated in the bottom of Figure 1, our relation attention module takes a set of proposal features as an input and outputs the enhanced representations with relation information for each proposal. These proposals can be adjacent or distant from each other. Specifically, our proposed module first projects all proposal features into a subspace, and then captures their relations via a pair-wise relation function. We then fuse information from all proposals to construct the enhanced features according to the learned relations. We call it *relation attention mechanism*, which in spirit is similar to the self-attention mechanism [24]. Our relation attention module is flexible and can be embedded in most of the existing networks because of the following properties: 1) no extra supervision is required because we do not need to define any constraint for what relations should be learned; 2) the relation attention module is designed in-place to keep the dimensions of input and output the same; and 3) the network with the relation attention module can be trained in an end-to-end manner. In this paper, we evaluate the effectiveness and generality of the proposed relation attention module on several existing temporal action localization methods. Using Structured Segment Networks [20], we achieve the state-of-the-art results on the Thumos14 dataset (43.7) and show significant improvements on the ActivityNet1.3 dataset.

The main contributions of this paper are as follows.

- We propose a relation attention module that effectively exploits the relation between video proposals and that can be embedded into current action localization algorithms with few modifications.
- We evaluate the effectiveness of adding relation information between proposals and show significant improvements compared to baselines on the Thumos14 and ActivityNet1.3 datasets for temporal action localization.

II. Related Work

A. Temporal Action Localization

Early works first generate proposals by sliding windows and then classify them using hand-crafted features [25]. Recently, with the development of deep learning [26]–[29] in image and video analysis, a great progress has been achieved in temporal action localization [30]–[37]. Some approaches generate a series of frame-level or clip-level action scores throughout an video and then collect temporal regions with continuous high scores as predicted results [30]–[32]. They suffer using long-range temporal information to predict the score for each unit. Recently, with the popularity of region-based detectors in the image domain [17], many approaches for temporal action localization leverage the "proposal+classification" paradigm, which first generates a set of proposals with potential action instances, and then recognizes them [33]–[36]. However, these methods rely on the quality of the generated proposals as well as the capability of classifiers. These methods also cannot be trained end-to-end to optimize proposals generation network and classification network simultaneously. Inspired by the success of Faster RCNN [38], some approaches tackle this issue by merging proposal and classification networks into an end-to-end trainable network [21], [22], [39]–[41]. Our proposed relation attention module is easy to embed in the architecture with proposals recognition stage, capturing the relation between proposals to assist action recognition.

B. Exploiting Contextual Information in Videos

The quality of proposal features is critical for action recognition and location regression. Many approaches attend to exploit contextual information to build a stronger video representation. [19], [20] extend beyond the start and end points of proposals by half of the proposals duration for proposal classification and location regression. [21] applies context feature to proposal generation stage. [22] uses context information for proposals ranking. [42] makes use of context information through constructing a feature bank, which is a collection of the features from each time step in the video. [43] uses predicted future information for online action detection. Although these approaches construct more powerful representation by exploiting contextual information, they ignore relation information between proposals, which is critical for temporal action localization task.

C. Object Relation in Image

Contextual information is also of interest in object detection. Before deep learning was commonly used, many methods leveraged the relation between objects in the post-processing stage to rescore detected objects [44], [45]. Some approaches also used more specific features, such as size and location, to capture object relations [45], [46]. When using deep learning, some methods model contextual information or object relations through sequential reasoning [47]–[49]. These methods are compatible to current state-of-the-art networks, although they are not designed in-place. The relation network in [50] designs an in-place module to explore object relations. However, in the video understanding domain, few methods explore

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

the effectiveness of leveraging relation between proposals. To the best of our knowledge, we are the first to verify the effectiveness of using proposal relation for temporal action localization task.

D. Attention Model

An attention mechanism has been leveraged in various tasks [24], [51]–[56]. [51] uses an attention model to aggregate a feature map from different scale inputs, improving semantic segmentation performance with small objects. Squeeze-and-Excitation Networks [52] fuse channel-wise features for image classification and object detection using an attention model. [24] uses a self-attention mechanism for language translation, and [53] proposes a non-local block to capture pixel-wise contextual information. DANet [54] makes use of both channel-wise and spatial attention to explore relation attention for scene segmentation. PSA [55] simultaneously considers semantic and location information between two pixels to aggregate contextual information. Our relation attention module learns a proposal-wise attention map to capture relative information for temporal action localization task.

III. Proposed Method

A. Motivations

Most methods of temporal action localization involve two stages: generating proposals and recognizing them. As discussed in *INTRODUCTION*, the relation between proposals is critical for recognition. However, most existing methods process proposals individually, neglecting relation information.

Therefore, we aim to design a module to capture relations among proposals, allowing the network to seek information from other proposals automatically and boost classification performance. We design our module with reference to the self-attention mechanism based on its success in solving dependency between words in machine translation [24]. We call the proposed module as *relation attention module* (RAM).

B. Relation Attention Module

We now illustrate the relation attention module more formally. Let $\boldsymbol{P} = \{p_k = (p_k^s, p_k^e)\}_{k=1}^K$ denote the proposal subset of one video, where K is the number of proposals, p_k^s and p_k^e are starting and ending points of the k^{th} proposal respectively. For the k^{th} proposal, we obtain the corresponding features $\boldsymbol{f}_k$ through a feature extractor and thus obtain the feature set $\boldsymbol{F} = \{\boldsymbol{f}_k\}_{k=1}^K$.

Given the input feature set $\boldsymbol{F}$, the output features of the relation attention module with respect to the k^{th} input features is computed as:

$$\boldsymbol{f}_k^R = \sum_{j=1}^{K} r(\boldsymbol{f}_k, \boldsymbol{f}_j) g(\boldsymbol{f}_j). \tag{1}$$

The function $r(\cdot)$ takes a pair of features as an input and outputs a scalar, representing the pair-wise relation weight. The function $g(\cdot)$ transforms the input features to the embedding subspace, and j is the index enumerating all input features.

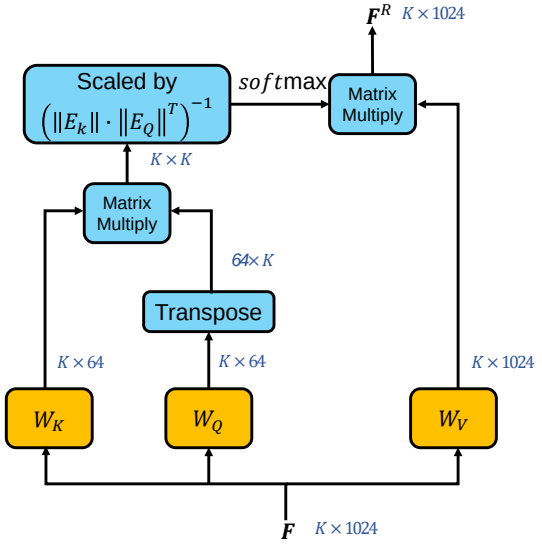

Fig. 2. The computation of relation attention module in *Sim-Cos* form with 64 dimension embedding. The number of features in $\boldsymbol{F}$ is K, and the dimension is 1024. W_K, W_Q and W_V transform the features into another subspace and are implemented as a 1×1 convolution. $||\cdot||$ calculates the l_2 norm of each row in the matrix. The softmax operation is performed on each row. The *Sim-Dot* and *Sim-Dot_Scale* instantiations can be performed by changing the *Scale* operation.

The output features for the k^{th} proposal can be viewed as the weighted average of all input proposal features in the subspace.

Following non-local neural network [53], function $g(\cdot)$ is simply designed as $g(\boldsymbol{f}_j) = W_V \boldsymbol{f}_j$, where W_V works as a linear embedding matrix, being implemented as a 1×1 convolution. We keep the embedding dimension the same as the input features. The pair-wise relation function $r(\cdot)$ is the key component of our proposed relation attention module and will be discussed next.

C. Pair-wise Relation Function

In this subsection, we explore several choices of the relation function and provide detailed descriptions.

Similarity. Inspired by "scaled dot-product attention" in [24], we use the similarity between two features followed by a softmax operation to exploit their relations. Specifically, we have:

$$r(\boldsymbol{f}_k, \boldsymbol{f}_j) = \frac{e^{\mathcal{S}(\boldsymbol{f}_k, \boldsymbol{f}_j)}}{\sum_{t=1}^{K} e^{\mathcal{S}(\boldsymbol{f}_k, \boldsymbol{f}_t)}}, \tag{2}$$

where $\mathcal{S}(\cdot)$ measures the similarity. Here, we formulate the function $\mathcal{S}(\cdot)$ as:

$$\mathcal{S}(\boldsymbol{f}_k, \boldsymbol{f}_j) = C \cdot \left[(W_Q \boldsymbol{f}_k)^T \cdot (W_K \boldsymbol{f}_j)\right], \tag{3}$$

where C is the scale factor; W_Q and W_K are two matrices transforming input features into two subspace with dimension d. In this paper, we choose multiple solutions for selecting C and explore their differences empirically. 1) When $C = [||W_Q \boldsymbol{f}_k|| \cdot ||W_K \boldsymbol{f}_j||]^{-1}$, $\mathcal{S}$ is the cosine similarity (*Sim-Cos*). 2) If C is set to 1, then $\mathcal{S}$ is the general dot product of the two embedding feature vectors (*Sim-Dot*). And Equation (1) becomes the "embedded Gaussian" form in non-local neural

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

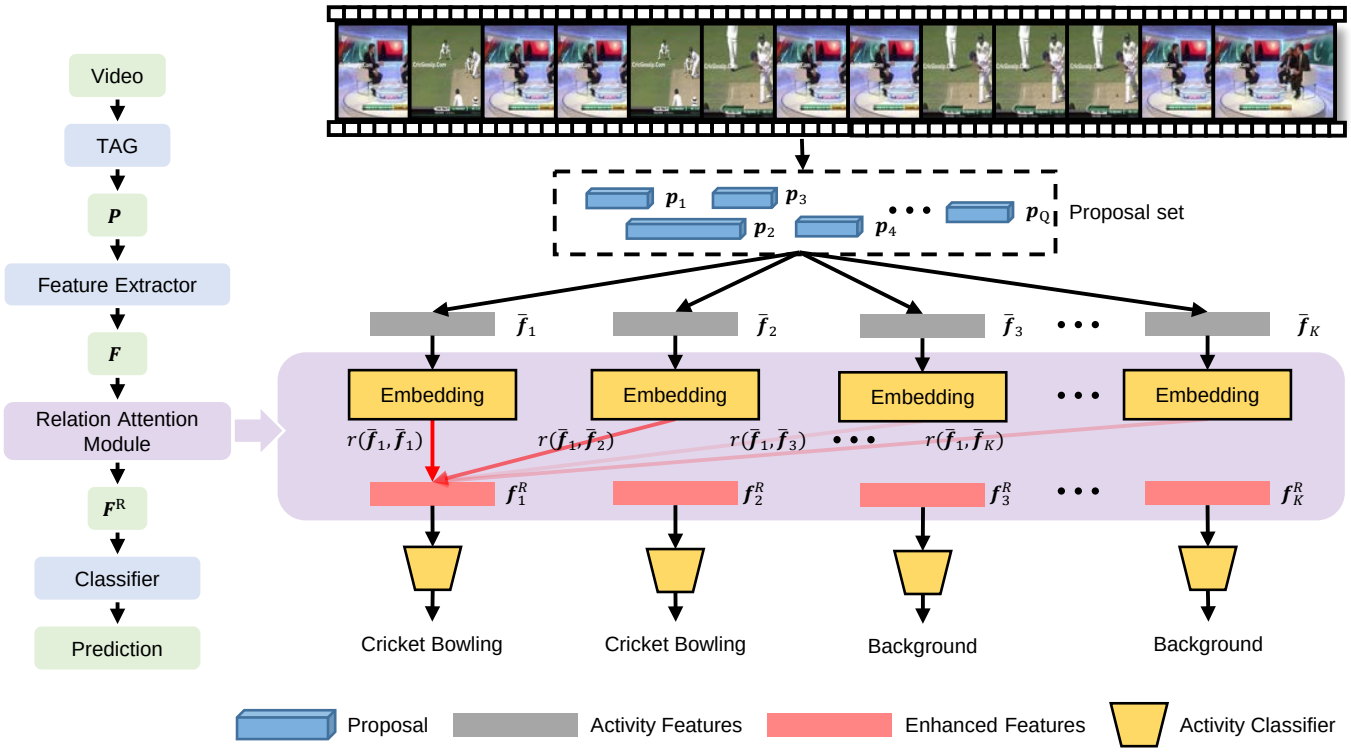

Fig. 3. The network architecture of SSN with our relation attention module for temporal action localization. For the relation attention module, we only show the process of obtaining the first enhanced features $\boldsymbol{f}_1^R$. The process of STPP, completeness classifier and location regressor are not shown.

networks. 3) When $C = \frac{1}{\sqrt{d}}$, the similarity function is the same as the self-attention mechanism in [24] (*Sim-Dot_Scale*).

Relation-FC. In addition to similarity, we can also use a fully-connected (*fc*) layer to instantiate function $\mathcal{S}(\cdot)$, and we call it *Relation-FC*. Specifically, two input features are concatenated in the subspace, followed by a *fc* layer with a scalar output together with the ReLU activation function [57]. Function $\mathcal{S}(\cdot)$ is defined as:

$$\mathcal{S}(W_Q \boldsymbol{f}_k, W_K \boldsymbol{f}_j) = \mathrm{ReLU}(\mathbf{w}_S \cdot [\mathrm{W_Q f_k}, \mathrm{W_K f_j}]), \quad (4)$$

where $[\cdot,\cdot]$ denotes the concatenation operation. Here, the relations between input features are modeled by a learnable vector $\mathbf{w}_S$.

All operations of the module can be implemented using basic operators. Its computation flow chart is shown in Figure 2. Next, we will illustrate how to use the relation attention module for temporal action localization.

D. Temporal Action Localization with RAM

Our module is designed in-place, which makes it flexible and easily integrated into most of the existing methods with few modifications. Based on the two-stage temporal action localization paradigm, we embed the proposed relation attention module into the Structured Segment Network (SSN) [20], a popular method for temporal action localization with good performance. Figure 3 shows the network architecture of SSN with the relation attention module. We first introduce the SSN pipeline and then describe how to embed the relation attention module into it.

In the first stage (proposal generation), the SSN generates a proposal set $\boldsymbol{P} = \{\boldsymbol{p}_q\}_{q=1}^{Q}$ by using the temporal actionness grouping (TAG) algorithm, which finds continuous temporal regions with high actionness scores to serve as proposals, where Q is the number of generated proposals. Several frames are selected uniformly from each proposal to construct activity feature set $\boldsymbol{F}$ through a feature extractor. The SSN also doubles the span of each proposal to involve the contextual information, leading to augmented proposal set $\boldsymbol{P'}$, and uses Structured Temporal Pyramid Pooling (STPP) to build the augmented feature set $\boldsymbol{F'}$. After extracting features, an activity classifier predicts the action category of each proposal using the activity features. A location regressor and a completeness classifier then take the augmented features as inputs, regress the relative changes from the proposal to the target action instance, and judge whether the proposal contains a complete action instance. Both the classifier and regressor are fully-connected layers. The activity and completeness classifiers are trained with cross-entropy loss, and the regression term is trained with L_1 loss.

We do not change the SSN pipeline but embed the relation attention module before the activity classifier to explore the relation between activity features. Specifically, in each iteration, we first randomly select K corresponding features from $\boldsymbol{F}$ and $\boldsymbol{F'}$ to build two subsets $\bar{\boldsymbol{F}} = \{\bar{\boldsymbol{f}}_k\}_{k=1}^{K}$ and $\bar{\boldsymbol{F}}'$. The relation attention module takes $\bar{\boldsymbol{F}}$ as input and outputs a collection of enhanced features $\boldsymbol{F}^R = \{\boldsymbol{f}_k^R\}_{k=1}^{K}$ using Equation (1). During training, considering the limitation of GPU memory, we select 8 proposals for each video at one iteration. In the testing phase, a variable number of proposals are selected

and their performances are discussed in the *EXPERIMENTS* section. Training details are shown in Algorithm 1.

Figure 3 does not show the process of STPP, the completeness classifier and location regressor because they are not necessary for the common two-stage temporal action localization paradigm. Embedding the relation attention module does not change the activity classifier but enhances its input features with information from various proposals. Thus, the relation attention module could also be embedded in any two-stage frameworks with proposals and classifiers.

E. Efficient Inference with RAM

In [20], the authors use a technique to reduce the inference time in the SSN, avoiding the redundant computations when extracting features for the same frames when inferring overlapped proposals. With this technique, the authors only need to extract features and calculate the classification and regression outputs for all sampled frames first, and then pool over these outputs to get the results for each proposal. We argue that the proposed relation attention module is compatible with this technique, and adding RAM does not add extra computational loads. The core operation of the relation attention module is described in Equation (1). During testing, similar to SSN, we first extract the feature set $\boldsymbol{V} = \{v\}_{t=1}^{T}$ for T uniformly sampled frames across the whole video. In Equation (1), the proposal features $\boldsymbol{f}$ are obtained by mean pooling over several frame features within proposal region r and can be represented as $\mathbb{E}_{t\sim r}[\boldsymbol{v_t}]$. Function $g(\boldsymbol{f})$ indicates linear embedding of $\boldsymbol{f}$. Assuming that W_e is a linear embedding matrix, we have:

$$g(\boldsymbol{f}) = W_e \cdot \boldsymbol{f} = W_e \cdot \mathbb{E}_{t\sim r}[\boldsymbol{v_t}] = \mathbb{E}_{t\sim r}[W_e \cdot \boldsymbol{v_t}]. \quad (5)$$

Equation (5) suggests that the proposal embedding features can be built by mean pooling over embedded frame features within the proposal region r. For the overlapped frames from two proposals, we do not need to extract these embedded frame features repeatedly but just pool over the extracted embedded frame features to build the embedded proposal features. With embedded features for each proposal, we can easily obtain the relation scores through the pair-wise relation function and then construct the enhanced features $\boldsymbol{f}^R$ through weighted average pooling. The final classification predictions are obtained from a *fc* layer upon $\boldsymbol{f}^R$. Empirically, adding RAM only incurs additional 0.06 seconds in inference time per video on average (from 0.78 to 0.84 seconds), which is negligible.

IV. EXPERIMENTS

In this section, we first describe datasets and evaluation metrics. Then, we embed RAM into several existing architectures to evaluate its effectiveness and generality, followed by an ablation study on RAM. We also evaluate the compatibility of RAM and visualize the results.

A. Datasets

The **Thumos14** [58] dataset contains 101 categories video and is composed of four parts: training, validation, testing and a background set. Each set includes 13320, 1010, 1574 and 2500 videos, respectively. Following the common setting in [58], we use 200 videos in the validation set for training, and 213 videos in the testing set for evaluation. The temporal action localization task of the THUMOS14 dataset is challenging because the video contains a large proportion of background information, and each video contains more than one action instance from one or multiple classes.

The **ActivityNet** [59] dataset is a standard benchmark for action recognition and localization tasks. We evaluate our method on the ActivityNet release 1.3, which has 200 activity classes and contains approximately 10,000 videos for training, 5,000 videos for validation, and 5,000 videos for testing. On average, each video contains 1.65 action instances from one or multiple classes. Per a standard practice, we train our model on training videos and test on validation videos.

Algorithm 1 Training details of SSN with RAM

Input: Training video set

```
generate proposal sets P and P' using TAG
extract feature sets F and F'
while not converges do
    select K features randomly to build F̄ and F̄'
    for f̄_k ∈ F̄ do
        construct f_k^R using Equation (1)
    end for
    predict action category from F^R
    predict completeness and boundary regression from F̄'
end while
```

Output: Trained model

B. Implementation Details

Training details. We train the SSN architecture with RAM in an end-to-end manner using 4 Nvidia TITAN X GPUs. The total training epochs are set to 450 and 70 for Thumos14 and ActivetyNet datasets, respectively. The model is updated by SGD with batch size 128 and momentum 0.9. We set the learning rate to 0.001 for both RGB and optical flow networks. By default, RAM is instantiated in *Sim-Cos* manner. The dimension d in the pair-wise relation function is set to 64. The number of proposals K selected for RAM is set to 8 in the training phase and set to 40% of the total number of proposals for each video in the testing phase.

Evaluation metrics. For quantitative evaluation, we use the mean Average Precision (mAP) as the comparison metric. Following the conventional evaluation set-ups, we report the mAP at different IoU thresholds. A prediction proposal is correct if it selects the same category as ground-truth and its temporal IoU with this ground-truth instance is larger than the IoU threshold. On the Thumos14 dataset, we choose IoU thresholds from [0.1, 0.2, 0.3, 0.4, 0.5], and mAP at 0.5 IoU is used for measuring the performance. On the ActivityNet1.3 dataset, the average mAP at the IoU thresholds [0.5:0.05:0.95] is used for evaluation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

TABLE I
IMPROVEMENT AFTER ADDING RAM UPON SEVERAL TEMPORAL ACTION LOCALIZATION ARCHITECTURES ON THUMOS14 TESTING SET. THE VALUE IN ( ) IS RELATIVE IMPROVEMENT.

Baseline	mAP@IoU=0.5(%)		
	Baseline	Baseline+RAM	Gain
SSN(RGB)	18.28	19.95	+1.67 (9.14%)
SSN(Flow)	23.30	28.34	+5.04 (21.63%)
SSN(RGB+Flow)	29.80	31.92	+2.12 (7.11%)
CBR	31.00	32.03	+1.03 (3.32%)
R-C3D	28.90	31.86	+2.96 (10.24%)

TABLE II
IMPROVEMENT AFTER ADDING RAM IN DIFFERENT PROPOSAL SETS ON RGB+FLOW SSN MODEL. WE USE IMAGENET PRE-TRAINED INCEPTION-V3 ARCHITECTURE AS BACKBONE NETWORK.

	mAP@IoU=0.5(%)		
	SSN	SSN+RAM	Gain
TAG	29.80	31.92	+2.12 (7.11%)
BSN	32.28	36.32	+4.04 (12.52%)
SW	25.79	29.10	+3.31 (12.83%)

TABLE III
IMPROVEMENT AFTER ADDING RAM ON DIFFERENT BACKBONE NETWORKS ON RGB+FLOW SSN MODEL. "IN" AND "KIN"MEAN MODEL PRE-TRAINED ON IMAGENET AND KINETICS DATASETS RESPECTIVELY. WE USE TAG PROPOSAL SET FOR TRAINING AND TESTING.

	mAP@IoU=0.5(%)		
	SSN	SSN+RAM	Gain
BN-Inception (IN)	27.06	30.23	+3.17 (11.71%)
Inception-v3 (IN)	29.80	31.92	+2.12 (7.11%)
BN-Inception (KIN)	31.58	34.67	+3.09 (9.78%)
Inception-v3 (KIN)	33.15	35.33	+2.18 (6.57%)

TABLE IV
IMPROVEMENT AFTER ADDING RAM INSTANTIATED IN DIFFERENT WAYS ON RGB SSN MODEL. WE USE KINETICS PRE-TRAINED BN-INCEPTION ARCHITECTURE AS BACKBONE NETWORK AND TAG PROPOSAL SET.

Model	mAP@IoU=0.5(%)	Gain
SSN baseline	22.07	-
Sim-Cos	24.16	**+2.09 (9.47%)**
Sim-Dot	23.04	+0.97 (4.3%)
Sim-Dot_Scale	23.42	+1.35 (6.12%)
Relation-FC	23.94	+1.87 (8.47%)

C. Effectiveness and Generality of RAM

To evaluate the effectiveness and generality of RAM, we embed it into the SSN, and other two temporal action localization methods (CBR [39] and R-C3D [40]). To determine whether RAM is robust to the quality of proposals and features, we also experiment with various proposal sets and different feature extractor backbone networks using the SSN architecture. We use the mAP at IoU 0.5 with the Thumos14 dataset as an evaluation metric in this subsection.

During training, K is set to 8 for the SSN and CBR, and 128 for R-C3D, considering the GPU memory limitation. In the testing, K is maintained at 40% of the total number of proposals for each video. A more detailed ablation study of RAM settings is shown in the next subsection.

Performance Beyond SSN. In [20], the SSN model is trained and tested on TAG proposals using ImageNet pre-trained Inception-v3 networks [60]. We follow this method to create fair comparisons and report results on both RGB and flow modalities in Table I. With the help of RAM, performance is boosted from 29.80 to 31.92. The gain in Flow is much larger than that in RGB, highlighting the advantages of RAM when processing the temporal information.

Performance Beyond CBR. The CBR model first constructs proposal features and then uses two *fc* layers to classify a proposal and regress its boundary in a cascaded manner. Before the final classification and coordinate regression layer in the detection stage, we use RAM to model relations among all input proposal features. Following [39], we use two-stream features and unit-level offsets. The results in Table I show that adding RAM yields +1.03 mAP improvement.

Performance Beyond R-C3D. The R-C3D model contains a proposal subnet that generates proposals and a classification subnet that predicts activity labels and regresses boundary. We embed RAM into the classification subnet and capture relations to enhance its features before the last activity classification *fc* layer. We follow all settings used in [40]. The results in Table I show that RAM yields a +2.96 mAP improvement.

Different types of proposals. Different algorithms generate proposal sets with different qualities and distributions. Table II shows performances using three different proposal sets, including TAG, BSN and SW. BSN indicates the proposal set generated by a boundary sensitive network [61], and we choose all generated proposals for training and the top 200 proposals with the highest scores from each video for testing, following the settings used in [61]. We also train our model using proposal set generated by the temporal sliding windows (SW) algorithm, where the length of sliding windows varies from 16, 32, 64, 128, 256, 512 frames with 75% overlap.

The results are shown in Table II. RAM yields consistent gains for all three types of proposals, indicating that RAM can capture useful relation information from proposal sets in various qualities. All proposal sets contain a certain proportion of background proposal (62.45%, 26.27% and 20.44% for SW, TAG and BSN respectively). We observe that background proposals are also used by RAM to enhance action proposal features. More detailed discussions of this topic are shown in Section IV-G.

Different types of backbone networks. The feature extractor with different backbone networks generates proposal features in different qualities. Table III shows the results on the BN-Inception [62] and Inception-v3 networks, both of which are pre-trained on the ImageNet or Kinetics datasets. Our module boosts performance consistently, meaning that our module does not rely on the proposal features from a specific network architecture. Without RAM, the Inception-v3 network outperforms the BN-Inception network by approximately 2% consistently (29.80 vs. 27.06 and 33.15 vs. 31.58) because of its deeper architecture. However, with RAM, the BN-Inception network outperforms the raw Inception-v3 network (30.23 vs. 29.80 and 34.67 vs. 33.15) by considering relation information.

D. Ablation Study

We first evaluate the performance of RAM with different pair-wise relation function, followed by an exploration of how

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

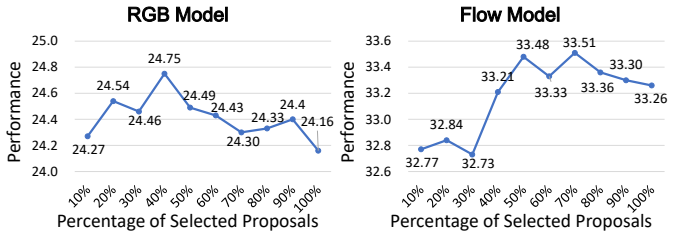

Fig. 4. Performance on SSN model with different percentages of selected proposals, measured by mAP@IoU=0.5. We use Kinetics pre-trained BN-Inception architecture as backbone network and TAG proposal set.

TABLE V
COMPARISON OF THE NUMBER OF LEARNABLE PARAMETERS AND THE PERFORMANCE GAINS ON RGB SSN MODEL. WE USE KINETICS PRE-TRAINED BN-INCEPTION ARCHITECTURE AS BACKBONE NETWORK AND TAG PROPOSAL SET.

Method	Parameters	mAP@IoU=0.5(%)
SSN	10.48 M	22.07
SSN+*fc*	11.53 M	22.32 (+0.25)
SSN+2*fc*	12.58 M	22.62 (+0.55)
SSN+RAM	11.66 M	**24.75 (+2.68)**

the number of related proposals affects performance. Then, we evaluate the RAM on both classification and regression stage, together with the effect of the incorporated parameters and the comparison between RAM and non-local block. After that, we explore the improvement in classification accuracy.

All ablation studies are implemented with the SSN structure using a TAG proposal set and a Kinetics pre-trained BN-Inception backbone network for relatively light training burden. We train the RGB model and report the mAP at IoU 0.5 on Thumos14 dataset for evaluation.

Choice of Pair-wise Relation Function. We investigate the effects of different pair-wise relation functions in our method. We use all proposals to calculate relations. The results are shown in Table IV. Considerable improvements are shown in all cases, varying from 0.97 to 2.09. Specifically, *Sim-Dot* and *Sim-Dot_Scale* gain the smallest improvements despite they sharing similar forms with non-local operation [53] and self-attention [24]. Among all candidates, *Sim-Cos* exhibits the best performance gain because the scale factor of *Sim-Cos* scales the output of relation function within -1 to 1, which provides better normalization. Thus, in the following ablation studies, we choose *Sim-Cos* as the pair-wise relation function by default.

Modeling Relation between Different Numbers of Proposals. As described above, our module can accept an arbitrary number of proposals as inputs. During training, following the settings in SSN, we choose 8 proposals from each video to fit available GPU memory and thus, only exploit the relations between these 8 proposals. In testing, since more proposals could be considered, we conduct experiments to explore performance versus the number of proposals.

Figure 4 shows how performances varies with different percentages of selected proposals per video. For the RGB model, the mAP first increases as more proposals are used to calculate relations, indicating that other proposals provide abundant information to help understand actions. Performance peaks when 40% of proposals are considered. After that, the mAP decreases as more proposals are considered, perhaps due to the incorporation of irrelevant information from some proposals. Similar results are observed in the Flow model. The results suggest that selecting 40% of proposals achieves a balance between the performance and the computation cost, and we follow this setting in the following experiments.

TABLE F
PERFORMANCE OF APPLYING RAM TO CLASSIFICATION (CLS.) AND REGRESSION (REG.) STAGES IN SSN STRUCTURE. WE USE KINETICS PRE-TRAINED BN-INCEPTION ARCHITECTURE AS BACKBONE NETWORK AND TAG PROPOSAL SET.

Method	mAP@IoU=0.5	Gain
SSN	22.07	-
SSN+RAM (Cls.)	24.75	+2.68 (12.14%)
SSN+RAM (Reg.)	23.32	+1.25 (5.66%)
SSN+RAM (Cls.+Reg.)	25.46	+3.39 (15.36%)

TABLE VII
COMPARISON WITH STATE-OF-THE-ART METHODS ON THUMOS14 DATASET. "-" INDICATES THE RESULT IS NOT REPORTED IN THE PAPER.

Method	mAP@IoU(%)				
	0.1	0.2	0.3	0.4	0.5
Shou et al. [36]	-	-	40.1	29.4	23.3
Yuan et al. [31]	51.0	45.2	36.5	27.8	17.8
Buch et al. [63]	-	-	45.7	-	29.2
Gao et al. [39]	60.1	56.7	50.1	41.3	31.0
Hou et al. [64]	51.3	-	43.7	-	22.0
Dai et al. [22]	-	-	-	33.3	25.6
Gao et al. [21]	54.0	50.9	44.1	34.9	25.6
Xu et al. [40]	54.5	51.5	44.8	35.6	28.9
Chao et al. [19]	59.8	57.1	53.2	48.5	42.8
SSN_M	62.54	60.46	56.93	51.11	41.36
SSN_M+RAM (ours)	**65.42**	**63.08**	**58.83**	**52.66**	**43.67**

RAM for Proposal Regression. In order to evaluate the benefits of RAM for proposal regression, we incorporate the RAM before location regressor of the SSN architecture. In Table F, applying the RAM to regression stage results in 1.25 mAP gain. When the RAM is applied to both classification and regression stages, the performance gain is more significant, booting from 22.07 to 25.46. This demonstrates that the relationships between proposals captured by the RAM are beneficial to recognizing activities as well as to locating actions more precisely. In this paper, we only consider the influences from RAM on the classification stage because such stage is necessary for two-stage temporal action localization paradigm while the regression stage is not [33].

Extra Learnable Parameters. To determine whether improvement comes from the additional learnable parameters or from the incorporation of RAM, we build a deeper SSN baseline by replacing RAM with a *fc* layer. Specifically, before being fed into the activity classifier, each proposal activity feature undergoes one or two *fc* layers independently. These *fc* layers incorporate more learnable parameters but cannot capture relations between proposals. Table V shows the number of parameters in the SSN baseline and several variants with their performance. We use *Sim-Cos* instantiated RAM with 40% of total proposals for testing. RAM introduces 1.18 M more parameters into the SSN baseline while yielding more than a 12% relative improvement in performance. Conversely,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

TABLE VIII
IMPROVEMENT AFTER ADDING RAM ON ACTIVITYNET1.3 DATASET.

Proposal	Method	mAP@IoU 0.5	0.75	0.95	Average
TAG	SSN_M	33.55	19.22	1.64	19.46
	SSN_M+RAM	35.14	20.35	2.15	20.68
	Gain	+1.59	+1.13	+0.51	+1.22
BSN	SSN_M	33.80	21.52	2.50	20.91
	SSN_M+RAM	36.99	23.10	3.34	23.03
	Gain	+3.19	+1.58	+0.84	+2.21

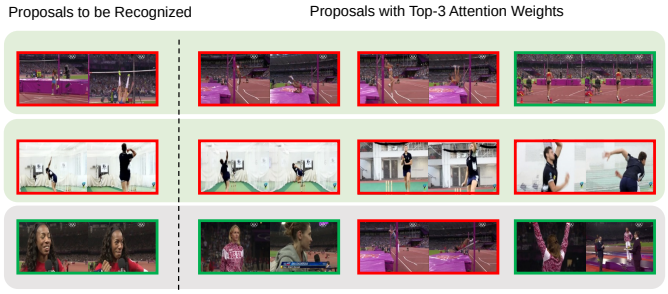

Fig. 5. Visualization of RAM. We visualize the testing results of *Sim-Cos* case in Table IV. Each proposal is represented by two uniformly selected frames. Proposals showed on the left of the dashed line are to be recognized and on the right side are the related proposals with top-3 scores. The red and green boxes represent action and background proposals respectively.

Fig. 6. Qualitative temporal action localization results on Thumos14 and ActivityNet1.3 datasets.

the "SSN+2*fc*" variant contains more parameters compared to the RAM variant but yields limited improvement. These comparisons indicate that the improvement due to RAM may not be attributed to the increased number of parameters only.

RAM vs. Non-local Block. RAM is similar to the self-attention method [24] for machine translation, where a specific output position consists of information from all positions of the input signal, while we use it for video understanding. The non-local neural network is also related to self-attention and is applicable to other domains in addition to machine translation. However, it models the relations between pixels of images or videos and thus captures the low-level features, which is useful for action classification. RAM instead focuses on the relations between high-level features (*i.e.*, proposal-level), which yields more semantic information.

To exploit the effectiveness of the non-local block on temporal action localization, we explicitly use it to capture pixel-level relations to enhance proposal features. Specifically, we use the Inception-v3 network as a feature extractor and add a non-local block after the "inception_4e" layer to capture relation information between different pixels. All other settings remain the same as the SSN. The experimental results show that, instead of increasing the performance, the mAP decreases from 22.07 to 21.03 and highlight the necessity of RAM. To the best of our knowledge, we are the first to leverage proposal relation for temporal action localization.

Importance of relationship information. In RAM, we enhance the proposal features through weighted averaging the randomly selected K proposals. The relation weight for each proposal is measured by a pair-wise function. We construct a simple baseline that assigns the same weight for each of K proposals. In this way, the mAP slightly increases from 22.07 to 22.47, which is significantly worse than the RAM (24.75). This is not surprising because although the context information from other proposals may be beneficial to recognizing actions, the relation weight for each proposal should be varied. The model should focus more on the relative content while the irrelevant background is supposed to have a small weight.

To calculate the weight for each proposal, one possible way is to concatenate all K proposal features and calculate K relation weights with a fully-connected (*fc*) layer. We construct another baseline, where we first transform K selected features into an embedding space with 512 dimensions via a learnable linear projection. Then, the concatenation of K embedded features undergoes a *fc* layer and outputs K relation weights. Such *fc* module incorporates a similar number of learnable parameters compared to RAM, while the performance gain is markedly lower than RAM (+0.46 vs. +2.68). The main difference between such fc module and our *Relation-FC* RAM is the capture of pair-wise relation weights. The results demonstrate that our attention relation module benefits from the design of the pair-wise relation function. Our module can also be adapted to different inputs using an arbitrary number of proposals, while the *fc* module requires a fixed number of proposals as inputs.

Improvement on Recognition. As described in the *INTRODUCTION* section, the second stage of the two-stage temporal action localization paradigm can be regarded as action recognition, which classifies the action proposals to their own classes and recognizes inaccurate proposals as background. RAM can construct enhanced features for recognition by capturing relation information from other proposals. We thus conduct experiments to evaluate the effectiveness of our proposed module at this process.

Specifically, we calculate the recognition accuracy of all TAG proposals. For the action proposals whose IoU with action instance is nonzero, we set the class of action instance with largest IoU as a target; for the other proposals, we expect the model to recognize them as background. Based on these settings, adding RAM increases recognition accuracy from 64.6% to 77.7%, demonstrating that the relative information captured by RAM is useful during recognition, which is critical to temporal action localization.

E. Comparisons with Other Methods

Current temporal action localization methods use certain techniques to achieve high performance. For example, TAL-Net in [19] uses "Inflated 3D ConvNet" [65] (I3D) features to boost performance. The effectiveness of the BSN proposal set for temporal action localization is also evaluated in [61]. We tend to evaluate the compatibility of RAM and these techniques. Thus, we modify the raw SSN by combining these two techniques with RAM. We denote the SSN using I3D features and BSN proposal set as modified SSN (SSN_M).

As shown in Table VII, the modified SSN boosts the performance of raw SSN and yields competitive results to [19] (41.36 vs. 42.8) in terms of mAP at IoU 0.5. Adding RAM into the SSN_M method further increases performance (from 41.36 to 43.67). With RAM, the SSN_M method outperforms all methods at all IoU thresholds, demonstrating that the proposed module is compatible with these techniques and can help the SSN achieve high performance.

F. Experiments on the ActivityNet Dataset

In addition to the Thumos14 dataset, we conduct experiments on the largest temporal action localization dataset, the ActivityNet1.3 dataset, to evaluate the generality of RAM. We use the modified SSN described above and evaluate both TAG and BSN proposals. As shown in Table VIII, adding RAM yields improvements in two proposal settings under all IoU thresholds and average mAP.

G. Visualization

We visualize the proposals with high attention weights in Figure 5. There are three proposals to be recognized, including two action proposals (top two on the left) and one background proposal (bottom one on the left). We show the proposals with the top-3 highest attention weights on the right side of the dashed line. We find that RAM tends to choose information from other action proposals to recognize action proposals. This is intuitive because different action instances belonging to the same class share similar characteristics. For the background proposals, our module learns to select other background proposals. Except for these intuitive findings, we also find that although the third highest related proposal in the first row only contains the background information (*e.g.*, the athletic field) related to the action of "high jumping", it still contributes to the recognition of that class. This interesting phenomenon indicates that our relation module can capture both actions and background information to enhance features. Additionally, in the third row, we note that the second related proposal spans a long duration, covering several discontinuous action instances, which is also chosen to be auxiliary information for recognition of background.

We also visualize qualitative localization results in Figure 6. The first two examples are from the Thumos14 dataset, and the last two are from the ActivityNet1.3 dataset. We find that with RAM, the model can classify action instances more accurately (*e.g.,* example 2) and localize instance boundaries more precisely (*e.g.,* example 4). Besides, the visual characteristics of action and background are similar across time in example 1. The model with RAM can localize the complete action instance, while the baseline fails.

V. Conclusions

In this paper, we proposed a simple relation attention mechanism to capture shared information among proposals to improve the temporal action localization performance. The proposed relation attention module can be embedded in most two-stage architectures with few modifications. We conducted extensive experiments to evaluate its effectiveness and generality. In the future, one possible modification is allowing the network to learn specific relation information based on different tasks.

References

[1] Z. Zhuang, M. Tan, B. Zhuang, J. Liu, Y. Guo, Q. Wu, J. Huang, and J. Zhu, "Discrimination-aware channel pruning for deep neural networks," in *NeurIPS*, 2018, pp. 875–886.

[2] Y. Guo, Q. Chen, J. Chen, Q. Wu, Q. Shi, and M. Tan, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Transactions on Multimedia*, vol. 21, pp. 2726–2737, 2019.

[3] D. Huang, P. Chen, R. Zeng, Q. Du, M. Tan, and C. Gan, "Location-aware graph convolutional networks for video question answering," in *AAAI*, 2020.

[4] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba, "Self-supervised moving vehicle tracking with stereo sound," in *ICCV*, 2019, pp. 7053–7062.

[5] G. Shen, W. Huang, C. Gan, M. Tan, J. Huang, W. Zhu, and B. Gong, "Facial image-to-video translation by a hidden affine transformation," in *ACM MM*, 2019, pp. 2505–2513.

[6] Y. Zhang, H. Chen, Y. Wei *et al.*, "From whole slide imaging to microscopy: Deep microscopy adaptation network for histopathology cancer image classification," in *MICCAI*, 2019.

[7] Y. Zhang, Y. Wei, P. Zhao, S. Niu *et al.*, "Collaborative unsupervised domain adaptation for medical image diagnosis," in *Medical Imaging meets NeurIPS*, 2019.

[8] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream cnn," *IEEE Transactions on Multimedia*, vol. 19, pp. 1510–1520, 2017.

[9] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length," *IEEE Transactions on Multimedia*, vol. 20, pp. 634–644, 2018.

[10] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang, "Social contextual recommendation," in *ACM MM*, 2012, pp. 45–54.

[11] Y. Yuan, T. Mei, and W. Zhu, "To find where you talk: Temporal sentence localization in video with attention based location regression," in *AAAI*, 2019, pp. 9159–9166.

[12] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Transactions on Multimedia*, vol. 9, pp. 257–267, 2007.

[13] B.-H. Chen and S.-C. Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Transactions on Multimedia*, vol. 16, pp. 837–847, 2014.

[14] W. Zhang, Q. J. Wu, G. Wang, and H. Yin, "An adaptive computational model for salient object detection," *IEEE Transactions on Multimedia*, vol. 12, pp. 300–316, 2010.

[15] K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE transactions on multimedia*, vol. 7, pp. 853–859, 2005.

[16] H. Sabirin and M. Kim, "Moving object detection and tracking using a spatio-temporal graph in h. 264/avc bitstreams for video surveillance," *IEEE Transactions on Multimedia*, vol. 14, pp. 657–668, 2012.

[17] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[18] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. G. Hauptmann, "Semi-supervised multiple feature analysis for action recognition," *IEEE Transactions on Multimedia*, vol. 16, pp. 289–298, 2014.

[19] Y. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster R-CNN architecture for temporal action localization," in *CVPR*, 2018, pp. 1130–1139.

[20] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *ICCV*, 2017, pp. 2933–2942.

[21] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "TURN TAP: temporal unit regression network for temporal action proposals," in *ICCV*, 2017, pp. 3648–3656.

[22] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *ICCV*, 2017, pp. 5727–5736.

[23] M. R. Boutell, J. Luo, and C. M. Brown, "Scene parsing using region-based generative models," *IEEE transactions on multimedia*, vol. 9, pp. 136–146, 2007.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 6000–6010.

[25] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *CVPR*, 2016, pp. 3093–3102.

[26] Y. Guo, Q. Wu, C. Deng, J. Chen, and M. Tan, "Double forward propagation for memorized batch normalization." in *AAAI*, 2018, pp. 3134–3141.

[27] Y. Guo, Y. Zheng, M. Tan, Q. Chen, J. Chen, P. Zhao, and J. Huang, "NAT: Neural architecture transformer for accurate and compact architectures," in *NeurIPS*, 2019.

[28] J. Cao, Y. Guo, Q. Wu, C. Shen, and M. Tan, "Adversarial learning with local coordinate coding," in *ICML*, 2018, pp. 706–714.

[29] J. Cao, L. Mo, Y. Zhang, K. Jia, C. Shen, and M. Tan, "Multi-marginal wasserstein gan," in *NeurIPS*, 2019.

[30] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *CVPR*, 2017, pp. 1003–1012.

[31] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, "Temporal action localization by structured maximal sums," in *CVPR*, 2017, pp. 3215–3223.

[32] R. Zeng, C. Gan, P. Chen, W. Huang, Q. Wu, and M. Tan, "Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization," *IEEE Transactions on Image Processing*, vol. 28, pp. 5797–5808, 2019.

[33] Z. Shou, D. Wang, and S. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016, pp. 1049–1058.

[34] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "DAPs: deep action proposals for action understanding," in *ECCV*, 2016, pp. 768–784.

[35] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: single-stream temporal action proposals," in *CVPR*, 2017, pp. 6373–6382.

[36] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S. Chang, "CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *CVPR*, 2017, pp. 1417–1426.

[37] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.

[38] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[39] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," in *BMVC*, 2017.

[40] H. Xu, A. Das, and K. Saenko, "R-C3D: region convolutional 3d network for temporal activity detection," in *ICCV*, 2017, pp. 5794–5803.

[41] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *ACM MM*, 2017, pp. 988–996.

[42] C. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krähenbühl, and R. B. Girshick, "Long-term feature banks for detailed video understanding," in *CVPR*, 2019, pp. 284–293.

[43] M. Xu, M. Gao, Y. Chen, L. S. Davis, and D. J. Crandall, "Temporal recurrent networks for online action detection," in *ICCV*, 2019, pp. 5532–5541.

[44] S. K. Divvala, D. Hoiem, J. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *CVPR*, 2009, pp. 1271–1278.

[45] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. L. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *CVPR*, 2014, pp. 891–898.

[46] M. J. Choi, A. Torralba, and A. S. Willsky, "A tree-based context model for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 240–252, 2012.

[47] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, "Attentive contexts for object detection," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 944–954, 2017.

[48] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *CVPR*, 2016, pp. 2325–2333.

[49] X. Chen and A. Gupta, "Spatial memory for context reasoning in object detection," in *ICCV*, 2017, pp. 4106–4116.

[50] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *CVPR*, 2018, pp. 3588–3597.

[51] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *CVPR*, 2016, pp. 3640–3649.

[52] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[53] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.

[54] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *CVPR*, 2019, pp. 3146–3154.

[55] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *ECCV*, 2018, pp. 270–286.

[56] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, "Visual grounding via accumulated attention," in *CVPR*, 2018, pp. 7746–7755.

[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[58] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[59] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015, pp. 961–970.

[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[61] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "BSN: boundary sensitive network for temporal action proposal generation," in *ECCV*, 2018, pp. 3–21.

[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[63] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *BMVC*, 2017.

[64] R. Hou, R. Sukthankar, and M. Shah, "Real-time temporal action localization in untrimmed videos by sub-action discovery," in *BMVC*, 2017.

[65] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017, pp. 4724–4733.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2019.2959977, IEEE Transactions on Multimedia

Peihao Chen received the B.E. degree in Automation Science and Engineering from South China University of Technology, China, in 2018. He is working toward the M.E. degree in the School of Software Engineering, South China University of Technology, China. His research interests include Deep Learning in Video and Audio Understanding.

Chuang Gan is currently a Researcher with the MIT-IBM Watson AI Lab. His research interests mainly include multi-modality learning for video understanding.

Guangyao Shen is currently a PH.D candidate in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include affective computing and video generation.

Wenbing Huang is now an assistant researcher (Postdoc level) at Tsinghua University, selected in the first group of Shuimu Tsinghua Scholar in 2019. Before this, he was a senior researcher at Tencent AI Lab during 2017 and 2019. He received the bachelor's degree in applied mathematics from Beihang University in 2012 and the Ph.D. degree in computer science and technology from Tsinghua University in 2017. His current research mainly lies in the areas of machine learning and computer vision, with particular focus on imitation learning, graph representation learning, and video understanding. He has published more than 20 peer-reviewed conference and journal papers, including the Proceedings of NeurIPS, ICML, CVPR, ICCV, ECCV, IJCAI, AAAI, WWW, ACMMM, and the IEEE Transactions on Image Processing and IEEE Transactions on Fuzzy Systems. He served as a pc member of IJCAI 2019-2020, AAAI 2019-2020, and a reviewer of NeurIPS 2019, CVPR 2019-2020, ICML 2019-2020, ICCV 2019, ECCV 2020, AISTATS 2019-2020, ACMMM 2019, WACV 2019-2020. He was selected in the list of Top Reviewers of NeurIPS 2019. He was a session chair of "Video: Events, Activities and Surveillance" of IJCAI 2019.

Runhao Zeng received the bachelor's degree in Automation Science and Engineering from South China University of Technology, Guangzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Software Engineering at South China University of Technology. His research interests include machine learning, deep learning and their applications in video understanding.

Mingkui Tan is currently a professor with the School of Software Engineering at South China University of Technology. He received his Bachelor Degree in Environmental Science and Engineering in 2006 and Master degree in Control Science and Engineering in 2009, both from Hunan University in Changsha, China. He received the Ph.D. degree in Computer Science from Nanyang Technological University, Singapore, in 2014. From 2014-2016, he worked as a Senior Research Associate on computer vision in the School of Computer Science, University of Adelaide, Australia. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.