Weakly-Supervised Video Re-Localization with Multiscale Attention Model

Yung-Han Huang,[1,2] **Kuang-Jui Hsu,**[1,2,3] **Shyh-Kang Jeng,**[2] **Yen-Yu Lin**[1,4]
[1]Academia Sinica, [2]National Taiwan University, [3]Qualcomm, [4]National Chiao Tung University
{R06946015,skjeng}@ntu.edu.tw, kuangjui@qti.qualcomm.com, lin@cs.nctu.edu.tw

Abstract

Video re-localization aims to localize a sub-sequence, called target segment, in an untrimmed reference video that is similar to a given query video. In this work, we propose an attention-based model to accomplish this task in a weakly supervised setting. Namely, we derive our CNN-based model without using the annotated locations of the target segments in reference videos. Our model contains three modules. First, it employs a pre-trained C3D network for feature extraction. Second, we design an attention mechanism to extract multiscale temporal features, which are then used to estimate the similarity between the query video and a reference video. Third, a localization layer detects where the target segment is in the reference video by determining whether each frame in the reference video is consistent with the query video. The resultant CNN model is derived based on the proposed co-attention loss which discriminatively separates the target segment from the reference video. This loss maximizes the similarity between the query video and the target segment while minimizing the similarity between the target segment and the rest of the reference video. Our model can be modified to fully supervised re-localization. Our method is evaluated on a public dataset and achieves the state-of-the-art performance under both weakly supervised and fully supervised settings.

Introduction

Video content understanding receives more and more attention since the number of videos in real world grows rapidly. One of the fundamental tasks for video analysis is action recognition (K. Simonyan and Zisserman 2014; Tran et al. 2015; Li et al. 2019; Hussein, Gavves, and Smeulders 2019; Girdhar et al. 2019; Diba1 et al. 2018). Most approaches for action recognition are developed to identify a human action inside a trimmed video. However, most real-world videos are untrimmed, which may hinder the advances in action recognition. Temporal action detection, i.e., action localization (Shou et al. 2017; Gao et al. 2017b; Zhao et al. 2017), has been proposed to work on untrimmed videos. Its goal is to separate the video clip of a specific action category from the rest. However, methods for temporal action detection are developed to detect the action segments of pre-defined categories. They often fail on actions of unseen classes.

Figure 1: Given a query video and a reference video, video re-localization aims to localize a segment, i.e., frames between indices 35 and 40, inside the reference that contains the same action category, i.e., biking, as the query.

To address this issue, Feng et al. (Feng et al. 2018) introduce a new task called *video re-localization*. As shown in Figure 1, the inputs of video re-localization are one trimmed query video and one untrimmed reference video, and the query is a short clip that a user is interested in. In the following, we assume the query is a human action unless otherwise specified. The goal of video re-localization is to search a segment in the reference such that this segment has the same semantic meaning as the query. The segment to be retrieved may not belong to a pre-defined category but the category the same as the query. Thus, the major difficulty of video re-localization lies in dealing with unseen categories. Despite effectiveness, video re-localization relies on frame-level action annotations. Annotating the exact locations of an action in the untrimmed video is time-consuming and expensive, and the high annotation cost of training data collection hinders the advances in video re-localization. To address this limitation, we propose a new approach trained with video-level, instead of frame-level, annotations, so the annotation cost is significantly reduced.

Inspired by the co-attention loss in (Hsu, Lin, and Chuang 2018) for co-segmentation, we extend this loss with the temporal cue for weakly supervised video re-localization. The co-attention loss considers merely spatial information, but the proposed temporal co-attention loss is designed to derive *convolutional neural networks* (CNNs) in a weakly super-

vised fashion. The resultant CNN model divides the frames of the reference video into two disjoint sets, the *target* and the *background* sets, in a discriminative way. That is, the CNN model aims at enhancing the similarity between the query and the target set while maximizing the dissimilarity between the target and the background sets.

Figure 2 illustrates the proposed model, which is composed of three components: a feature extractor, an attention module, and a localization predictor. Specifically, we employ a pre-trained C3D (Tran et al. 2015) to extract video features. The second component consists of two submodules, *multiscale attention* and *self-attention* modules. The former calculates the similarity between the query and the target set using the their features of arbitrary lengths, and thus better temporal information are better exploited. The latter, i.e., the self-attention module in (Vaswani et al. 2017), is adopted to encode context information of the reference video into the local features. We concatenate the similarity and the local features as the input of the localization predictor. Finally, the localization predictor can determine if each reference frame has the same semantic meaning as the query or not.

We make the following contributions in this work. First, to the best of our knowledge, this work makes the first attempt to develop a weakly supervised CNN-based model for video re-localization. Second, we propose temporal-based co-attention loss and a novel attention module to capture not only multiscale temporal structure across the different frames but also the local cues in a single frame. Third, our method is evaluated on the standard benchmark for video re-localization (Feng et al. 2018), and remarkably outperforms the state-of-the-art method (Feng et al. 2018).

Related work

Video Action Recognition. Video action recognition aims to identify the action category of a video. Recently, many effective methods for action recognition are developed based on deep learning to extract and fuse the spatial and temporal cues, such as the two-stream model (K. Simonyan and Zisserman 2014), C3D (Tran et al. 2015), CoST (Li et al. 2019), timeception (Hussein, Gavves, and Smeulders 2019), transformer network (Girdhar et al. 2019), and spatio-temporal channel correlation (Diba1 et al. 2018). Some existing approaches further integrate attention into action recognition to boost the performance via different forms, such as the attention-based gating or second order pooling (Girdhar et al. 2019; Girdhar and Ramanan 2017; Long et al. 2018; Xie et al. 2018), guidance from the human pose (Baradel, Wolf, and Mille 2017; 2018) and self-attention (Wang et al. 2018; Girdhar et al. 2019). Despite their effectiveness on action recognition on trimmed videos, they may fail on the untrimmed videos. Our method is developed for the untrimmed videos. Besides, different from the attention-based action recognition in a single video, we design the attention-based model to capture the same action clips across two videos.

Temporal Action Detection.

Given untrimmed training videos and the corresponding frame-level annotations with the predefined action categories, temporal action detection, aka action localization, learns a model to seek each action segment belonging to one of these categories. A widely used way for action localization (Shou et al. 2017; Gao et al. 2017b; Zhao et al. 2017) is to first detect potential video segments with the confidence scores, and then identify human actions inside the video having higher confidence scores. Recently, Lin et al. (Lin et al. 2018) propose to predict a set of the start and end points for an action clip instead of using time-consuming sliding window schemes or error-prone pre-defined intervals for video segment generation. Long et al. (Long et al. 2019) utilizes Gaussian kernels further take the temporal structure of action clips into account.

Nevertheless, it is difficult to collect the frame-level location annotations. To address this issue, weakly-supervised methods (Wang et al. 2017; Nguyen et al. 2018; Liu, Jiang, and Wang 2019) are proposed to work with video-level labels. However, they have two limitations. First, their learned models are not applicable to unseen action categories, which making these models less practical. Second, frame-level discriminative learning is not considered, which maximizes the difference between the action frames and the background frames. It helps better separate the target action clips from the background clips.

Action Localization with Sentence/Video. In (Gao et al. 2017a; Hendricks et al. 2017), a task called action localization with a query sentence is proposed. Given a video and a query sentence, this task aims to find the temporal boundary of the video clip, which best matches the text description. In (Chen et al. 2018), an attention-based method is proposed to preserve not only global temporal information but also local feature details. In (Zhang et al. 2019a), a single-shot feed-forward network with the iterative graph adjustment network cell is designed to unify the candidate clip encoding and temporal structural reasoning.

Instead of using the language sentence as the query, Feng et al. (Feng et al. 2018) propose a new task called *video re-localization* where a query video and a reference video are given. Different from sentence-based action localization, video re-localization targets at finding a subsequence of the reference video clip that best matches the query (or covers the same action as the query for action localization). Feng et al. (Feng et al. 2018) design a cross gated bilinear matching scheme upon a CNN-based model composed of three LSTM layers. In gated bilinear matching, complicated interactions between features of the query video and the reference video are exploited and captured. The information inside the query video is encoded as a single global feature which is a weighted sum of all features generated from the query frames. Therefore, the local temporal structure cannot be preserved. Furthermore, the method in (Feng et al. 2018) requires the frame-level location annotations that are difficult to collect. In this work, we propose a weakly supervised video re-localization method without using frame-level supervisory signals, and hence the annotation cost can be significantly reduced. In addition, we design a multiscale attention module to capture the local temporal structure, and can better discover the target action clip in the reference video.

It is worth mentioning that some existing papers extend-

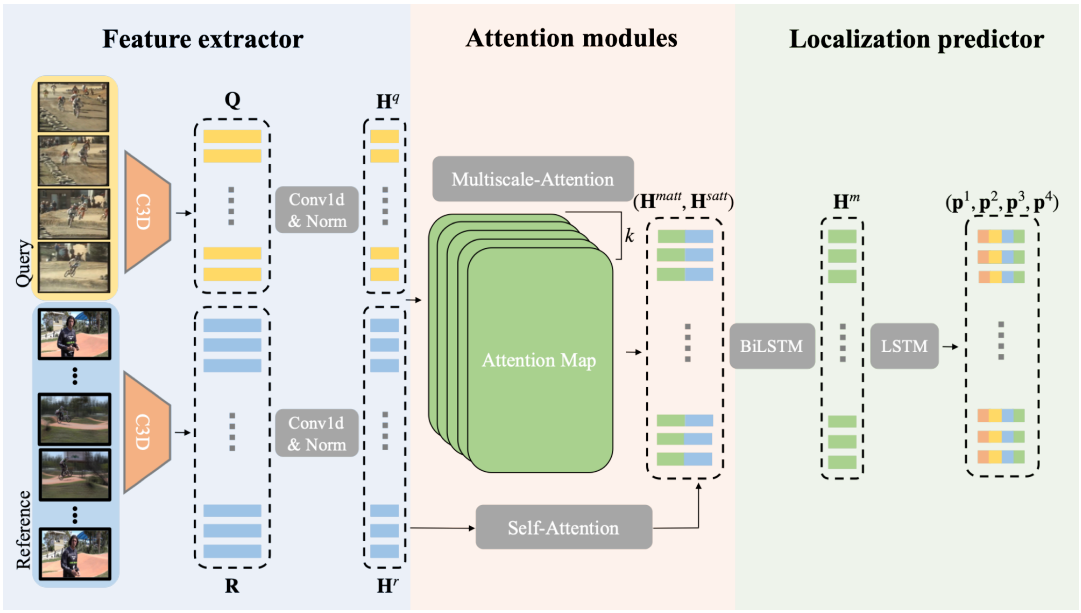


Figure 2: Our model is composed of three components: the feature extractor, attention module, and localization module. First, we employ a pre-trained C3D network to serve as the feature extractor. Second, we design a multiscale attention mechanism to compute the similarity between the query and the reference videos. We also introduce self-attention module at the same time to encode the context information into local features. Finally, the localization layer predicts if each frame of the reference video matches the query video.

ing the task of video re-localization in diverse ways. For example, Feng et al. (Feng et al. 2019) not only localize the temporal boundary of an action clip but also seek the spatial bounding box of that action at each frame. Zhang et al. (Zhang et al. 2019b) try to use a single query image to localize the relevant action clip in a reference video.

Approach

We introduce the proposed method in this section. The problem definition is first given. Then, we describe our attention module and introduce the deep network architecture in our method. Finally, two loss functions for training and the inference process at the testing stage are detailed.

Problem Definition

Video re-localization is applied to a pair of videos, including a query video $\mathbf{V}^Q = \{\mathbf{v}_n^Q\}_{n=1}^q$ and a reference video $\mathbf{V}^R = \{\mathbf{v}_n^R\}_{n=1}^r$, where $\mathbf{v}_n^Q$ and $\mathbf{v}_n^R$ are the n-th frame in the query video and the reference video, respectively, and q and r are the numbers of frames in the two videos respectively. For action localization, the query video is a trimmed action clip and the reference video is an untrimmed video which contains an action instance of the same category as the query video. Our goal is to seek the temporal boundary of an action instance $\{\mathbf{v}_n^R\}_{n=t_s}^{t_e}$ in $\mathbf{V}^R$ that best matches the query, where t_s and t_e are the start and the end timestamps, respectively. The frames between t_s and t_e are in the action region and the other frames belong to the background region. In the weakly-supervised condition, our model is derived without ground truth location information $\{t_s, t_e\}$.

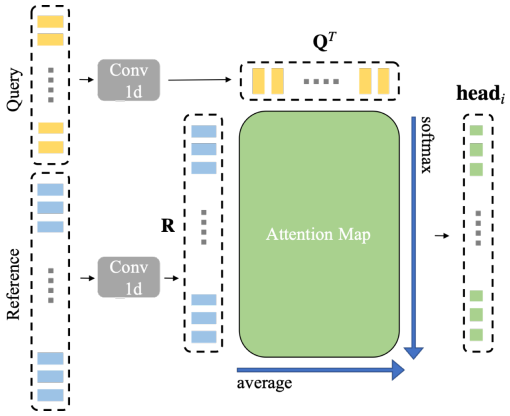


Figure 3: A "head" of multiscale attention module. We compute the outer product between the query features and reference features to obtain an attention map. A softmax function is then applied to each column of the attention map for normalization. Finally, we average each row of the attention map to obtain the attention score sequence.

Attention Module

Self-attention and multiscale attention modules are used in this work, and are detained as follows.

Self-Attention Module Transformer is proposed in (Vaswani et al. 2017), which is an attention model and is effective for machine translation. In addition to word sequences, this model is also applicable to video sequences.

In our method, its variant, the multi-head self attention module, is adopted and it can be formulated as follows:

$$f^{selfAtt}(\mathbf{X}) = Concat(\mathbf{head}_1, ..., \mathbf{head}_h), \quad (1)$$

$$\mathbf{head}_i = Attention(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V), \quad (2)$$

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}. \quad (3)$$

The input $\mathbf{X} \in \mathbb{R}^{l_v \times d_f}$ is a video with l_v frames, each of which is represented by d_f-dimensional features. Each *head* maps $\mathbf{X}$ into three components $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ by learning three matrices $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$. Each component and each matrix are of size $\mathbb{R}^{l_v \times d_k}$ and $\mathbb{R}^{d_f \times d_k}$, respectively. The three components correspond to the sets of queries, keys and values, respectively. Then dot-product attention is computed in (3), where $\frac{1}{\sqrt{d_k}}$ is used for scaling the dot product. The output of self-attention is the concatenation from all heads. It gather different features, since each head learns its weights in the process. Refer to (Vaswani et al. 2017) for the details.

Multiscale Attention Module Inspired by (Vaswani et al. 2017), we propose a multiscale attention module as illustrated in Figure 3. Given a query video $\mathbf{Q} \in \mathbb{R}^{l_q \times d_f}$ and a reference video $\mathbf{R} \in \mathbb{R}^{l_r \times d_f}$, we calculate the attention score between a frame of the reference video and all frames in the query video by

$$Attention(\mathbf{R}, \mathbf{Q}) = avgpool(softmax(\frac{\mathbf{R}\mathbf{Q}^T}{\sqrt{d_k}})). \quad (4)$$

In (4), we compute the outer product between $\mathbf{R}$ and $\mathbf{Q}^T$ to yield an attention map, which records the pairwise similarity between each pair of frames, one from query video and one from the reference video. We use $\frac{1}{\sqrt{d_k}}$ for scaling the dot product. Then, we apply a softmax function to normalize the score between each query frame to all frames of the reference video. After this operation, each column of the attention map becomes a probability distribution. Finally, we average each row of the attention map to obtain the single attention score for each frame of the reference video, which indicates if this frame is well matched by the query.

We also adopt the multi-head strategy in this attention module by

$$f^{multiAtt}(\mathbf{R}, \mathbf{Q}) = Concat(\mathbf{head}_1, ..., \mathbf{head}_k), \quad (5)$$

$$\mathbf{head}_i = Attention(f_i^{conv1}(\mathbf{R}), f_i^{conv2}(\mathbf{Q})). \quad (6)$$

Different from the self-attention module using weight matrices for projection, we employ 1d convolutional layers, f_i^{conv1} and f_i^{conv2}, to encode temporally structured information within a surrounding window. Considering the large variations on action lengths, we use different kernel sizes and dilation rates in each head. Higher dilation rates are with larger receptive fields such that we can compute attention scores based on features of different temporal scales, which are crucial to the performance of video re-localization. The final output $f^{multiAtt}(\mathbf{R}, \mathbf{Q}) \in \mathbb{R}^{l_r \times k}$ is the concatenation of all heads.

Network Architecture

Figure 2 gives an overview of the proposed network architecture, which is composed of three components: feature extractor, attention modules and localization predictor. We detail these components below.

Feature Extractor Given a query video $\mathbf{V}^Q$ and a reference video $\mathbf{V}^R$, we use a pre-trained C3D network to extract their features. The extracted features are respectively denoted by $\mathbf{Q} \in \mathbb{R}^{l_q \times d_f}$ and $\mathbf{R} \in \mathbb{R}^{l_r \times d_f}$, where l_q and l_r are the numbers of frames in the two videos respectively while d_f is the dimension of a feature vector. Then, we employ 1d convolutional layers followed by layer normalization (Ba, Kiros, and Hinton 2016) to map $\mathbf{Q}$ and $\mathbf{R}$ into $\mathbf{H}^q \in \mathbb{R}^{l_q \times d_h}$ and $\mathbf{H}^r \in \mathbb{R}^{l_r \times d_h}$ respectively, i.e.,

$$\mathbf{H}^q = f_q^{conv}(\mathbf{Q}) \text{ and } \mathbf{H}^r = f_r^{conv}(\mathbf{R}). \quad (7)$$

Note that f_q^{conv} and f_r^{conv} in (7) do not share weights. We use two independent layers for $\mathbf{Q}$ and $\mathbf{R}$, since their properties are different. For example, the feature vectors in $\mathbf{Q}$ are more similar to each other than those in $\mathbf{R}$ because the query video contains a trimmed action clip while the reference video may contain multiple action and background segments.

Attention Modules After extracting the features of videos, we employ two attention modules described above to enhance the signals of the action region and suppress signals of background regions in a reference video.

The first step is to apply the proposed multiscale attention module to capture the interactions between the query video and a reference video, namely

$$\mathbf{H}^{matt} = f^{multiAtt}(\mathbf{H}^r, \mathbf{H}^q), \quad (8)$$

where $\mathbf{H}^{matt} \in \mathbb{R}^{l_r \times k}$ is composed of k sequences of attention score. These attention score sequences are computed based on convolutional layers with different kernel sizes and dilation rates. Therefore, $\mathbf{H}^{matt}$ brings information from various temporal scales.

Then, we apply the self-attention layer to the reference video, i.e.,

$$\mathbf{H}^{satt} = f^{selfAtt}(\mathbf{H}^r). \quad (9)$$

Note that the query video is not taken into account in (9) where we focus on the relationships between frames in the reference video. We fuse the context information according to the attention weights at each timestamp in the reference video. The output $\mathbf{H}^{satt} \in \mathbb{R}^{l_r \times hd_k}$ encodes self-attention between frames of the reference video, and hence is effective in separating the action part from the background part of the reference video. Therefore, we concatenate $\mathbf{H}^{matt}$ and $\mathbf{H}^{satt}$ and use them as the input to the localization layer.

Localization Predictor Following (Feng et al. 2018), we use a localization predictor to classify the frames of the reference video into four categories: starting point, ending point, action region, and background region (out of action region). The exact action boundary is then inferred via post-processing, which will be specified later.

The localization predictor consists of two LSTM layers. The first LSTM is bidirectional and is applied to encode the global information into each local feature, i.e.,

$$\mathbf{h}_i^m = BiLSTM(Concat(\mathbf{h}_i^{matt}, \mathbf{h}_i^{satt}), \mathbf{h}_{i-1}^m), \quad (10)$$

where $\mathbf{h}_i^{matt} \in \mathbb{R}^{1\times k}$ and $\mathbf{h}_i^{satt} \in \mathbb{R}^{1\times 2d_h}$ are the i-th rows of $\mathbf{H}^{matt}$ and $\mathbf{H}^{satt}$ respectively. They represent features at timestamp i while $\mathbf{h}_i^m$ is the hidden state.

The second LSTM layer accompanied with a softmax function classifies each frame i of the reference video:

$$\mathbf{p}_i = softmax(\mathbf{h}_i^l \mathbf{W}^l + \mathbf{b}^l), \quad (11)$$

$$\mathbf{h}_i^l = LSTM(\mathbf{h}_i^m, \mathbf{h}_{i-1}^l), \quad (12)$$

where $\mathbf{h}_i^l \in \mathbb{R}^{1\times d_l}$ is the hidden state of the LSTM. $\mathbf{W}^l \in \mathbb{R}^{d_l\times 4}$ and $\mathbf{b}^l \in \mathbb{R}^{1\times 4}$ are trainable parameters. The output $\mathbf{p}_i \in \mathbb{R}^{1\times 4}$ represents the probabilities of being a starting point(p_i^1), an ending point (p_i^2), action (p_i^3), and background (p_i^4) at the i-th frame of the reference video. And we denote the probability sequences over the reference video as $\mathbf{p}^j = [p_1^j, ..., p_{l_r}^j]$ for $j \in \{1, 2, 3, 4\}$.

Objective Function

The co-attention loss is proposed by (Hsu, Lin, and Chuang 2018) and is used for unsupervised object co-segmentation. It aims to segment the object region without pixel-level annotation. We adopt the co-attention loss to train our model under the weakly supervised setting. Our model can also be modified for fully supervised video re-localization by updating the weighted cross entropy loss.

Co-Attention Loss Video clips belonging to the same action category are supposed to have similar representations in the feature space. That is, the distance between their features should be small. On the other hand, the distance of features from an action video clip to background clips or other clips of different categories should be larger. Utilizing this property, the co-attention loss can guide the model to separate the action part and background part of a reference video. More importantly, it does not need frame-level location annotations, and calculates only the distance between features to detect the plausible action region in the reference video.

To begin with, we average the extracted features of the query video to represent the query, i.e.,

$$\mathbf{q}^a = \frac{1}{l_q}\sum_{i=1}^{l_q} \mathbf{q}_i \in \mathbb{R}^{1\times d_f}, \quad (13)$$

where $\mathbf{q}_i \in \mathbb{R}^{1\times d_f}$ is the i-th row of $\mathbf{Q}$ and the feature of query video at timestamp i. Similarly, we have $\mathbf{r}_i \in \mathbb{R}^{1\times d_f}$, the i-th row of $\mathbf{R}$, for the reference video.

Then, we normalize the probability sequence $\mathbf{p}^j \in \mathbb{R}^{l_r\times 1}$, which is predicted by the localization layer, along the time axis to obtain the normalized action probability sequence

$$g_i^j = \frac{p_i^j}{\sum_{i=1}^{l_r} p_i^j} \in \mathbb{R}^{l_r\times 1}. \quad (14)$$

Here we use the normalized action probability sequence $\mathbf{g}^3$ to weight the reference video features as $\mathbf{r}^a \in \mathbb{R}^{1\times d_f}$, which represents the action region features of the reference video. Similarly, we use the normalized background probability sequence $\mathbf{g}^4$ as weights to yield $\mathbf{r}^b$, which represents the background region features of the reference video, i.e.,

$$\mathbf{r}^a = \sum_{i=1}^{l_r} \mathbf{r}_i \cdot g_i^3 \text{ and } \mathbf{r}^b = \sum_{i=1}^{l_r} \mathbf{r}_i \cdot g_i^4. \quad (15)$$

The co-attention loss is defined by

$$loss^c = \frac{\exp(-d^+)}{\exp(-d^+) + \exp(-d^-)}, \quad (16)$$

$$d^+ = \cos(\mathbf{q}^a, \mathbf{r}^a), \quad (17)$$

$$d^- = \alpha \cdot \cos(\mathbf{q}^a, \mathbf{r}^b) + (1-\alpha)\cdot \cos(\mathbf{r}^a, \mathbf{r}^b), \quad (18)$$

where $\cos(\cdot,\cdot)$ denotes the cosine distance and α is a hyperparameter, which is used to balance the two terms in the equation. Minimizing this loss $loss^c$ will minimize the distance between $\mathbf{q}^a$ and $\mathbf{r}^a$. At the same time, it also maximizes the distance between $\mathbf{r}^a$ and $\mathbf{r}^b$ as well as the distance between $\mathbf{q}^a$ and $\mathbf{r}^b$. This loss can guide the model to separate the action region from the background region.

Weighted Cross Entropy Loss Our model can be generalized to fully supervised video re-localization by using the weighted cross entropy loss (Feng et al. 2018) where the frame-level location annotation are exploited. The weighted cross entropy loss $loss^w$ is defined by

$$loss^w = -\frac{1}{l_r}\sum_{i=1}^{l_r} w_i \sum_{n=1}^{4} y_i^n log(p_i^n), \quad (19)$$

where y_i^n is the ground truth and w_i is the weight to make model pay more attention on the boundary. Because there are only one starting point and one ending point in a reference video, we need to give heavier weight at boundary positions. Following (Feng et al. 2018), we set w_i to 10 if $y_i^1 + y_i^2 > 0$, which means that it is a starting point or an ending point at timestamp$i$; Otherwise, we set w_i to 1.

Inference

By taking a pair of videos, a query video and a reference video, as input to the learned model, we obtain four sequences of probabilities. We need to figure out the exact starting and ending timestamps, s and e. In the fully supervised setting, we follow (Feng et al. 2018) and make the prediction by

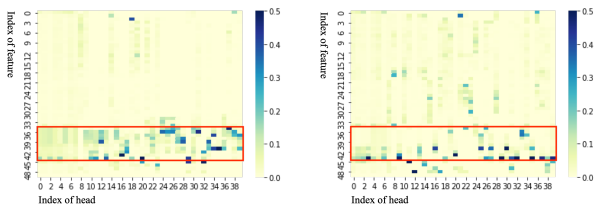
$$(s, e) = \arg\max_{s,e} p_s^1 p_e^2 (\prod_{i=s}^{e} p_i^3)^{\frac{1}{e-s+1}} \quad (20)$$

Finally, we will choose the segment with the largest joint probability as the prediction of the starting and the ending timestamps.

In the weakly-supervised setting, we do not have the location annotation in the training stage. Without the cross entropy loss, the network cannot predict $\mathbf{p}^1$ and $\mathbf{p}^2$. Therefore, we directly use $\mathbf{p}^3$ to find s and e. Specifically, we average $\mathbf{p}^3$ into $\bar{p}^3$ and use it as a threshold. We seek the longest segment whose p_i^3 are all above the threshold as the final prediction.

number of heads	kernel size	dilation rate	number of kernels
8	1	1	32
8	3	1	32
8	3	2	32
8	3	3	32
8	3	4	32

Table 1: We show the parameters used in the multiscale attention module.



(a) fully supervised (b) weakly supervised

Figure 4: We visualize the "heads" of the multiscale attention modules. There are 40 columns in each figure that represent the outputs of the 40 heads. The parameter setting of each head can be referred to Table 1. There are 49 rows to represent the 49 timestampes in the reference video. The red box frames the action region of it.

Experimental Results

In this section, we evaluate our method on the benchmark dataset. We first introduce the dataset and the evaluation metric for video re-localization. Then, we give the details of our implementation and introduce the methods for comparison. The experimental results and ablation study under weakly supervision and fully supervised settings are exhibited. Finally, we visualize the output of multiscale attention module for gaining insight into the quantitive results.

Dataset and Evaluation Metric

Dataset The dataset used for video re-localization is collected in (Feng et al. 2018) from ActivityNet (Heilbron et al. 2015), which is a large-scale action localization dataset with segment-level action annotations. The new collected dataset has 9,400 untrimmed videos, and each video contains only one action instance. The dataset is split into three disjoint sets including the training, validation and testing sets. The training set contains 160 classes and totally 7,593 videos. There are 7,593 videos of 160 classes in the training set, 978 videos of 20 classes in the validation set and 829 videos of 20 classes in the testing set. For the training of the fully supervised method, we randomly sample two videos in the same action class. Then, we crop one of them into a trimmed action segment as a query video, and the other one is considered as as the reference video. However, when we train the weakly supervised model, the training dataset is split into two parts in advance, one part for the query videos and the other for the reference videos, to avoid the usage of the groundtruth.

Evaluation metric We use the evaluation metric in the action localization task to evaluate the proposed method. Following (Feng et al. 2018), we calculate the top-1 mean average precision (mAP) under the four temporal IoU thresholds including 0.5, 0.6, 0.7, 0.8 and 0.9. The average over these four mAPs is also calculated to the result evaluation.

Implementation Details

For each video, the features are first extracted with the pretrained C3D network provided by the ActivityNet Challenge 2016. Each feature is extracted over 16 frames without the temporal overlap. Then, with these extracted features, we use the principal component analysis method (PCA) for the dimension reduction to deduce the computational cost. The dimensions of both features, d_q and d_r, are 500. The dimensions of the hidden state d_h and d_l are both set as 128. In the self attention layer, we use two heads, i.e. $h = 2$, and set d_k as 32. In the multiscale attention layer, we use 40 heads, i.e. $k = 40$. The details of the convolutional layer parameters of each head are summarized in Table 1. We adopt dropout before and after the $BiLSTM$ in the localization predictor, and the dropout rates are set as 0.4 and 0.2, respectively. The model is optimized by an Adam solver with a batch size of 128. The initial learning rate is set as 0.001 and decreased by 10 per 200 iterations. In the inference stage, the maximum number frames in the predicted segment are set as 1024 frames for the fully supervision setting, but it is not limited in the weakly supervision setting.

For the weakly-supervised setting, the co-attention loss is adopted during the training. Before calculating the co-attention loss, we first use the predicted action region, to generate an action mask and a background mask. Then, we apply these mask on the $\mathbf{p}^3$ and $\mathbf{p}^4$, respectively. We found this mask is beneficial to the weakly-supervised training. More specifically, we use the co-attention loss without the masks at the first 100 iterations, then optimize the model via the co-attention loss with the masks at the remaining iterations.

Competing Methods

We summarize the competing methods we adopt here.

- **Random.** We randomly pick a starting and an ending time within the length of the reference video as predictions.
- **Frame-level baseline (Feng et al. 2018).** This baseline needs neither localization labels nor training. It calculates the distance between each pair of normalized features in the query video and the reference video. It make prediction based on average distance of the segment.
- **Video-level baseline (Feng et al. 2018).** The prediction is obtained by using a LSTM to encode each video segment in the reference video into a vector, and finding the segment most similar to that of the query video. The weights in LSTM are updated by minimizing the triplet loss.
- **Statistical prior baseline.** We first normalize the length of each video to 1, and then compute the average starting and end points in the training set, which are at 0.3265 and 0.6509, respectively. Finally, we apply this prior knowledge to predict the testing set.

mAP @1	0.5	0.6	0.7	0.8	0.9	Avg.
Random	16.2	11.0	5.4	2.9	1.2	7.3
FL Baseline	18.8	13.9	9.6	5.0	2.3	9.9
ours						
base model	34.4	25.1	17.0	9.8	3.6	18.0
+multiscale	34.7	26.1	20.0	11.2	3.7	19.2
+selfatt	35.0	28.5	21.0	10.2	3.6	19.7
full model	**36.5**	**30.6**	**21.2**	**11.4**	**3.8**	**20.7**

Table 2: Performance of the proposed weakly supervised model on the testing set.

- **SST (Buch et al. 2017).** SST is a model designed for action proposal. Given a video, it would finds all of possible action segments and assign a confidence score to each one. For video re-localization task, the segment with the highest confidence will be chosen as the final prediction.
- **CGBM model (Feng et al. 2018).** It is a model with three layers of LSTM. In addition, they design cross gating and bilinear matching mechanisms to capture the interactions between query video features and reference video features.

Comparison with Weakly Supervised Methods

Table 2 shows the results of our model trained under weakly supervision setting. We compare our model with other methods trained without frame level location annotation. From the table, we can observe that our model exceeds FL Baseline (Frame-level baseline) by about ten percent on average top-1 mAP. It even surpasses some fully supervised method in Table 4.

Ablation study is also conducted to evaluate the attention modules in our model. For our base model, we remove the self-attention features ($\mathbf{H}^{satt}$) and set all convolutional layers in the multiscale module with kernel size 3 and dilation rate 1. We have tried to use 1D convolutional layers with kernel size 1, but it didn't work. By observing the attention maps in the module, we find that the outputs of convolutional layers with kernel size 1 are "flatter" than the layers with bigger kernel size. This may lead to the model hard to separate the action region and the background regions. The results in Table 2 shows that both of the multiscale and the self attention mechanisms improved the performance.

We also tried to find the best ratio α for co-attention loss. The experiment results are presented in Table 3. It shows that in addition to considering the relation between the reference video and the query video, pushing the distance between the action region feature and the background region feature in the reference video is also important. The model achieves the best performance as $\alpha = 0.3$.

Comparison with Fully Supervised Methods

Our method can easily be modified for the fully supervision setting by training with the weighted cross entropy loss. In Table 4, we show the performance of our model under the fully supervised setting and make a comparison with other methods trained under the same condition. In the table, our model is better than the CGBM model (Feng et al. 2018), which is the current state-of-the-art method, only using the cross entropy loss without the co-attention loss. The performance gain is attributed to the use of attention modules. Our attention mechanism keeps temporal information more completely by using convolutional layers with different kernel sizes and dilation rates. We also report the result by our full model. With the co-attention loss, our model can get 1 more percent performance gain. It shows that the co-attention loss is also helpful under the fully supervised setting.

mAP @1	0.5	0.6	0.7	0.8	0.9	Avg.
$\alpha = 1$	34.4	25.4	18.5	10.5	3.2	18.4
$\alpha = 0.7$	35.4	27.6	19.9	10.8	4.8	19.7
$\alpha = 0.5$	34.6	28.0	21.0	12.2	4.9	20.1
$\alpha = 0.3$	36.5	30.6	21.2	11.4	3.8	20.7
$\alpha = 0$	32.7	26.7	19.9	11.2	5.2	19.0

Table 3: Performance with different values of α.

mAP @1	0.5	0.6	0.7	0.8	0.9	Avg.
Stat. prior	25.4	16.5	2.3	2.3	1.2	10.7
VL Baseline	24.3	17.4	12.0	5.9	2.2	12.4
SST	33.2	24.7	17.2	7.8	2.7	17.1
CGBM	43.5	35.1	27.3	16.2	6.5	25.7
ours						
$loss^w$	**45.3**	**37.6**	**27.7**	**17.0**	**8.3**	**27.2**
$loss^w + loss^c$	**46.5**	**37.8**	**29.7**	**18.0**	**8.7**	**28.2**

Table 4: Performance of the proposed fully supervised model on the testing set.

Visualization

An advantage of our model is that its working can be interpreted via the attention mechanism. Thus, we here visualize the outputs of the multiscale attention modules in Figure 4 with the input data sampled from the testing set. Figures 4(a) and 4(b) are the outputs of our method under fully supervision and weakly supervision settings, respectively. From the figure, we find that heads exactly detect on differences of features. For the outputs of the fully supervised attention module, values in the red box are obviously higher than values outside. It shows that the cross entropy loss can guide the attention module to attend on the action region more precisely than simply using the co-attention loss only. In addition, we can think of a head output as a probability distribution and then observe that the output of heads with kernel size of 1 (head index of 1 to 8) is more flat. On the other hand, heads with larger vision fields have more clear peak.

Conclusions

In this work, we have introduced a new task of weakly supervised video re-localization. To this end, we proposed a model with the co-attention loss which utilizes the distance of features to guide the model separating the action region from the background region in an unsupervised manner. To preserve the local temporal structure, we design the multi-

scale attention module which has multiple heads to extract features from various temporal scales. Finally, the experimental results show that our model achieves the state-of-the-art performance on the benchmark dataset in both weakly supervised and fully supervised settings.

Acknowledgments

This work was funded in part by Qualcomm through a Taiwan University Research Collaboration Project. This work was also supported in part by Ministry of Science and Technology (MOST) under grants MOST 107-2628-E-001-005-MY3 and MOST 108-2634-F-007-009.

References

Ba, J.; Kiros, J.; and Hinton, G. 2016. Layer normalization. *arXiv*.

Baradel, F.; Wolf, C.; and Mille, J. 2017. Human action recognition: Pose-based attention draws focus to hands. In *ICCV Workshop*.

Baradel, F.; Wolf, C.; and Mille, J. 2018. Human activity recognition with pose-driven attention to RGB. In *BMVC*.

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Niebles, J. C. 2017. SST: Single-stream temporal action proposals. In *CVPR*.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018. Temporally grounding natural sentence in video. In *EMNLP*.

Diba1, A.; Fayyaz, M.; Sharma, V.; Arzani, M.; Yousefzadeh, R.; Gall, J.; and Gool, L. V. 2018. Spatio-temporal channel correlation networks for action classification. In *ECCV*.

Feng, Y.; Ma, L.; Liu, W.; Zhang, T.; and Luo, J. 2018. Video re-localization. In *ECCV*.

Feng, Y.; Ma, L.; Liu, W.; and Luo, J. 2019. Spatio-temporal video re-localization by warp lstm. In *CVPR*.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017a. Tall: Temporal activity localization via language query. In *ICCV*.

Gao, J.; Yang, Z.; Chen, K.; Sun, C.; and Nevatia, R. 2017b. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*.

Girdhar, R., and Ramanan, D. 2017. Attentional pooling for action recognition. In *NeurIPS*.

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *CVPR*.

Heilbron, F.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*.

Hsu, K.-J.; Lin, Y.-Y.; and Chuang, Y.-Y. 2018. Co-attention CNNs for unsupervised object co-segmentation. In *IJCAI*.

Hussein, N.; Gavves, E.; and Smeulders, A. 2019. Timeception for complex action recognition. In *CVPR*.

K. Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*.

Li, C.; Zhong, Q.; Xie, D.; and Pu, S. 2019. Collaborative spatiotemporal feature learning for video action recognition. In *CVPR*.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*.

Liu, D.; Jiang, T.; and Wang, Y. 2019. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*.

Long, X.; Gan, C.; de Melo, G.; Wu, J.; Liu, X.; and Wen, S. 2018. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian temporal awareness networks for action localization. In *CVPR*.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, L.; Xiong, Y.; Lin, D.; and Gool, L. V. 2017. UntrimmedNets for weakly supervised action recognition and detection. In *CVPR*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*.

Zhang, D.; Dai, X.; Wang, X.; Wang, Y.-F.; and Davis, L. 2019a. MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*.

Zhang, Z.; Zhao, Z.; Lin, Z.; Song, J.; and Cai, D. 2019b. Localizing unseen activities in video via image query. In *IJCAI*.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*.