

Action Shuffling for Weakly Supervised Temporal Localization

Xiao-Yu Zhang, *Senior Member, IEEE*, Haichao Shi, Changsheng Li, *Member, IEEE*,
and Xinchu Shi

Abstract—Weakly supervised action localization is a challenging task with extensive applications, which aims to identify actions and the corresponding temporal intervals with only video-level annotations available. This paper analyzes the order-sensitive and location-insensitive properties of actions, and embodies them into a self-augmented learning framework to improve the weakly supervised action localization performance. To be specific, we propose a novel two-branch network architecture with intra/inter-action shuffling, referred to as ActShufNet. The intra-action shuffling branch lays out a self-supervised order prediction task to augment the video representation with inner-video relevance, whereas the inter-action shuffling branch imposes a reorganizing strategy on the existing action contents to augment the training set without resorting to any external resources. Furthermore, the global-local adversarial training is presented to enhance the model’s robustness to irrelevant noises. Extensive experiments are conducted on three benchmark datasets, and the results clearly demonstrate the efficacy of the proposed method.

Index Terms—Temporal Action Localization, Self-Supervised, Inter-Action, Intra-Action.

I. INTRODUCTION

TEMPORAL action localization is one of the most challenging tasks in video content understanding, which has attracted intensive attention in the community. Given an untrimmed video, action localization aims to identify the time intervals corresponding to the actions of interest. Remarkable progress has been made in the fully supervised scenario [1], [2], [3], [4], where frame-level annotations are indispensable. Unfortunately, the overwhelming labeling effort to obtain the detailed annotations renders the fully supervised methods inapplicable to large-scale video sets. This leads to the prevalence of weakly supervised paradigm [5], [6], [7], [8], [9], [10], [11], which only requires video-level annotations to deduce frame-level predictions. To date, various weakly supervised action localization methods have been put forward and the most recent advances manifest two striking trends. (1) **Action-background modeling**. As indicated by the latest works, explicitly modeling the action and background contents proves to be an effective way of representation learning [6], [7], [9], [8]. By learning the video-level action and background representations of a video separately, action localization performance can be

X.-Y. Zhang and H. Shi are with Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, 100093. H. Shi is also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China. (e-mail: zhangxiaoyu@iie.ac.cn; shihaichao@iie.ac.cn)

C. Li is with Beijing Institute of Technology, Beijing, China. (e-mail: changshengli507@163.com)

X. Shi is with Meituan Group, Beijing, China. (e-mail: shixinchu@meituan.com)

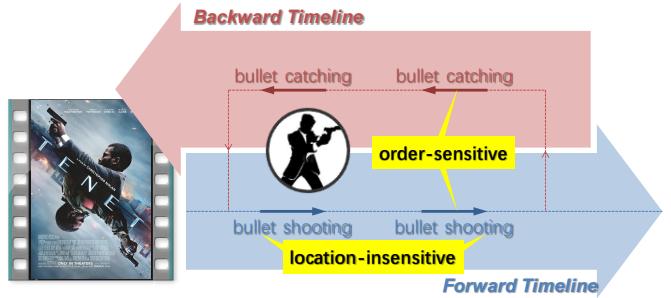


Fig. 1: Illustration of the order-sensitive and location-insensitive properties of actions.

improved collaboratively. However, the video-level modeling can only capture coarse-grained description. Less study has been directed on in-depth analysis of the intrinsic properties of actions. (2) **Exploration of external resources**. In order to make up for the limited information in weak supervision, it has become another trend to resort to external resources. Typically, the publicly available videos or generated pseudo videos with video-level or frame-level labels are leveraged as supplementary training data [12], [6], [13]. For all the performance gain achieved by this manner, new challenges have also emerged. For one thing, source-target adaptation between the original and auxiliary dataset is vital but difficult for robust knowledge transfer. For another, feature extraction of the new training videos lays extra burden on computational consumption. To address the above issues, this paper aims to reveal the properties of actions and embody them into the model to achieve effective and efficient weakly supervised action localization.

Tenet, a 2020 hit movie directed by Christopher Nolan, opens up a dramatic world with the art of “time inversion”, where the characters can go back in time. This paper does not intend to be a spoiler of the story. However, there are two interesting phenomena which enlighten us greatly in action analysis. In the movie, when going backward in time, actions become weird and hard to understand. In contrast, when a time point in the past has been reached via backward time travelling and the forward timeline is restored, actions get back to normal and seem to blend in well although the surrounding environments have changed drastically. This mind-blowing movie indicates two critical properties of actions illustrated in Fig. 1.

- On one hand, actions are *order-sensitive*. As we know, the dynamic motional properties of a video are reflected by the temporal relevance within ordered frames. Altering

the inner order of an action may significantly change its semantics. Particularly, in Tenet, bullet shooting in reversed order becomes bullet catching, which is an utterly different action.

- On the other hand, actions are *location-insensitive*. Compared to the dependence on inner order, an action is relatively independent on when it takes place. Taking actions of the same category in different time points is not likely to affect the underlying semantics, as long as the original inner order is retained.

Inspired by the order-sensitive and location-insensitive properties of actions, in this paper, we propose a novel weakly supervised action localization network architecture with intra/inter-action shuffling, referred to as ActShufNet. On top of the conventional attention-based action recognition and localization paradigm, we build a self-augmented learning model to achieve improved representative ability, without resorting to any external resources. Starting with the preliminarily segmented actions based on class-agnostic attentions, our model goes through two pipelines, i.e. the intra-action and inter-action shuffling. Intra-action shuffling randomly alters the inner order of an action and aims to restore its original order via a self-supervised task. In this way, the optimized representations are forced to capture the underlying inner relevance of actions, which facilitate the subsequent semantic deduction. Inter-action shuffling randomly picks actions of the same category and collectively creates new untrimmed videos which are naturally attached with the shared video-level labels. In this way, the training dataset can be arbitrarily expanded, and meanwhile more variety is included within each created video. To further enhance the model's discriminative ability between action and background, global-local adversarial training scheme is presented to achieve perturbation tolerable robust learning performance. The main contributions of our work are summarized as follows.

- We develop the intra/inter-action shuffling mechanism to fully exploit the order-sensitive and location-insensitive properties of actions and improve the model's representative ability. The model works in a self-augmented fashion, where no external resources are required.
- We design the global-local adversarial training scheme to enhance the model's robustness to irrelevant noises, with respect to video-level prediction and segment-level action-background discrimination.
- We lay out the network architecture to integrate separate modules into a unified framework, which is optimized in an end-to-end fashion. Extensive experiments on challenging untrimmed video datasets show promising results of ActShufNet over the state-of-the-arts.

The rest of this article is organized as follows. We review the related work in Section II and introduce the details of the proposed method in Section III. The results of experimental evaluation are reported in Section IV, followed by conclusions in Section V.

II. RELATED WORK

Action recognition is aimed to determine the categories of human actions in a trimmed video. Earlier methods extract

hand-crafted features, such as Improved Dense Trajectory (iDT) [14], [15], consisting of MBH, HOF and HOG features extracted along dense trajectories. Recently, with the development of deep learning, various learning based methods have been proposed, two-stream networks [16] learn both spatial and temporal features by operating network on single frame and stacked optical flow field respectively using 2D Convolutional Neural Network. C3D [17] uses 3D convolutional networks to capture both spatial and temporal information directly to learn discriminative features. I3D [18] is exploited to use a 3D version of Inception network [19] under the two-stream architecture. Wang et al. [20] developed a temporal segment network to perform space sparse sampling and fuse the temporal results. There are also having methods using recurrent neural networks to model temporal information, such as LSTM [21].

Temporal action localization is aimed to identify the temporal intervals which contain target actions. Previous works mainly focus on designing hand-crafted feature representations to classify the sliding windows [22]. Recently, fully supervised action localization methods leverage the ideology of object detection to obtain improved localization results. SSAD [23] utilizes the 1D temporal convolutional layers to directly detect action instances in untrimmed videos. SSN [1] proposes to utilize a structured temporal pyramid to model the temporal structure of the action instances. S-CNN [24] utilizes multi-stage CNNs to learn hierarchical feature representations. BSN [3] utilizes a multi-stage local to global fashion to generate temporal proposals. Besides the fully supervised methods, weakly supervised methods have also been extensively studied, which can be categorized into two classes. Top-down methods (*e.g.*, UntrimmedNets [25], W-TALC [5], 3C-Net [26], CMCS [6], BasNet [10]) learn a video-level classifier and then generate the frame activation score to localize actions. Top-down methods directly learn the temporal attention from videos and optimize the attention with video classification task. TSRNet [12] takes advantage of self-attention mechanism and transfer learning and integrates them to obtain precise temporal intervals in untrimmed videos. Autoloc [27] is proposed to train the boundary predictor with an outerinner-contrastive loss to directly predict the temporal boundary of each action instance. STPN [28] adds a sparse constraint to encourage the action sparsity. BM [7] penalizes the background features and proposes a clustering loss to separate actions and backgrounds. DGAM [9] proposes to model the class-agnostic frame-wise probability conditioned on the frame attention using conditional VAE.

Self-supervised learning is one type of techniques that learn representations by solving labourious annotation tasks, where the pseudo supervision signals can be obtained. Self-supervised learning for video analysis aims to learn the motion representations from the unlabeled data by solving the pre-text tasks. Wang et al. [29] exploits different self-supervised approaches to learn representations invariant to inter-instance and intra-instance variations among object patches, which are extracted from unlabeled videos using motion cues. In [30], the chronological order of frames in the video are exploited to learn a robust temporal features. Similarly, Luo and Wang

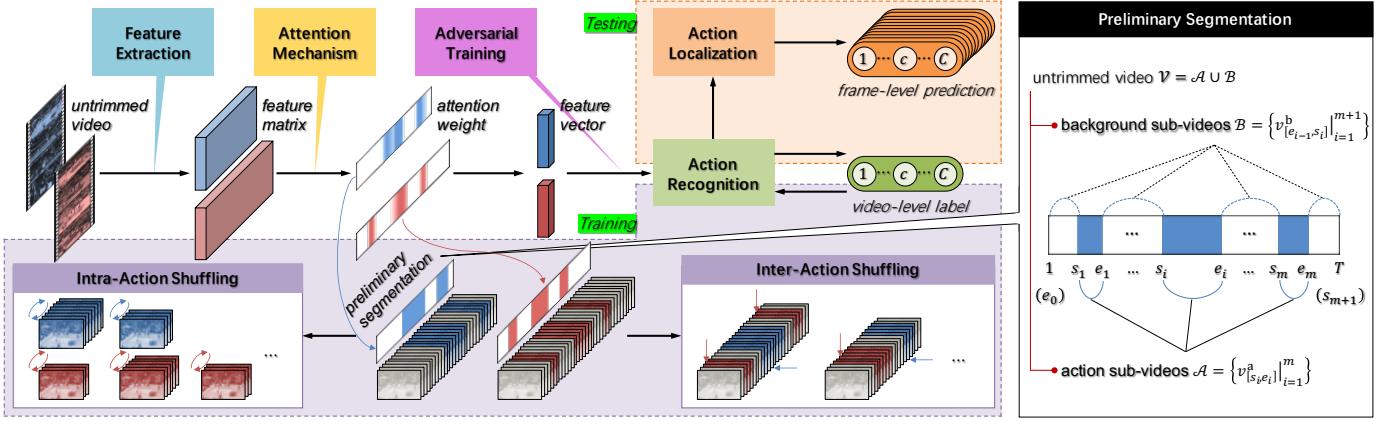


Fig. 2: Detailed framework of ActShufNet. In the training stage, ActShufNet starts from *Feature Extraction*. The extracted features are then fed into *Attention Mechanism* module to obtain compact feature representations. Then the action recognition task is optimized with an *Adversarial Training* scheme. In the testing stage, the localization results are obtained.

et al. [31], [32] propose to learn the video representation by predicting motion flows. Inspired by [30], we also explore the chronological order of action frames to learn precise features. To the best of our knowledge, this is the first attempt that integrates self-supervised learning with weakly supervised action localization.

Adversarial learning has been utilized in computer vision fields to reconstruct target features. In general, it is used to improve the robustness of model. Goodfellow et al. propose an adversarial method namely FGSM [33] and FGM [34], which make the direction of perturbation along the direction of gradient improvement. Madry et al. [35] propose to use projected gradient descent (PGD) to address the internal maximum problem. Since the seminal work by Goodfellow et al. [36] in 2014, a series of GAN family methods have been proposed for a wide variety of problems, which are based on adversarial process corresponding to a minimax two-player game. By means of adversarial learning, we construct the adversary between actions and backgrounds to discriminate them precisely.

III. PROPOSED METHOD

In this section, we present the framework of intra/inter-action shuffling, i.e., ActShufNet, which is illustrated in Fig. 2. As a weakly supervised learning model, ActShufNet learns from untrimmed videos and the corresponding video-level labels in the training stage, and predicts frame-level labels of untrimmed videos in the testing stage. For a video $\mathcal{V} = \{f_t\}_{t=1}^T$ of T frames/snippets, we follow the two-stream standard practice and extract the RGB or optical flow video features $\mathbf{X} = [\mathbf{x}_t]_{t=1}^T \in \mathbb{R}^{d \times T}$ with a pre-trained feature extraction model, where $\mathbf{x}_t \in \mathbb{R}^d$ is the feature vector of the t -th frame/snippet and d is the feature dimension. Without loss of generality, we use frame-wise feature extraction, though the proposed method is also applicable to snippet-wise features. The video-level label is denoted as $\mathbf{y} = [y_c]_{c=1}^{C+1} \in \mathbb{R}^{C+1}$, where C is the number of actions of interest and the $(C+1)$ -th class corresponds to background. Given \mathbf{X} , the model outputs the non-overlapping action instances as $\{(s_i, e_i, \mathbf{p}_i)\}_{i=1}^m$,

where s_i , e_i , and \mathbf{p}_i represent the start time, end time, and predictive class label of the i -th action instance respectively, and m is the number of identified action instances.

A. Attention-Based Representation Learning

Untrimmed videos of variable length T bring about variable-sized feature matrices, which are extremely inconvenient to process. Therefore, we leverage the attention-based mechanism to integrate the frame-level descriptions, and obtain fixed-sized compact representations.

The attention-based representation learning module aims at deriving attention vector $\boldsymbol{\lambda} = [\lambda_t]_{t=1}^T \in \mathbb{R}^T$ of an untrimmed video by optimizing the action recognition task. The attention weight $\lambda_t \in [0, 1]$ indicates the contribution of the t -th frame in identifying an action. Using $\boldsymbol{\lambda}$ to perform attention-weighted temporal average pooling over frames, we arrive at the fixed-sized action representation $\mathbf{x}_{[\text{start}, \text{end}]}^a \in \mathbb{R}^d$ of video segment $v_{[\text{start}, \text{end}]} = \{f_t\}_{t=\text{start}}^{\text{end}}$ between an arbitrary interval $[\text{start}, \text{end}]$ ($1 \leq \text{start} < \text{end} \leq T$) as follows.

$$\mathbf{x}_{[\text{start}, \text{end}]}^a = \frac{\sum_{t=\text{start}}^{\text{end}} \lambda_t \mathbf{x}_t}{\sum_{t=\text{start}}^{\text{end}} \lambda_t}. \quad (1)$$

Specifically for the entire video \mathcal{V} , the action feature is computed as $\mathbf{x}^a = \mathbf{x}_{[1, T]}^a$.

Similarly, $(1 - \lambda_t)$ can be regarded as the confidence that no actions are taking place in frame t and we can calculate the background representation $\mathbf{x}_{[\text{start}, \text{end}]}^b \in \mathbb{R}^d$ as follows.

$$\mathbf{x}_{[\text{start}, \text{end}]}^b = \frac{\sum_{t=1}^T (1 - \lambda_t) \mathbf{x}_t}{\sum_{t=1}^T (1 - \lambda_t)}. \quad (2)$$

The background feature for the entire video \mathcal{V} is $\mathbf{x}^b = \mathbf{x}_{[1, T]}^b$

For action recognition, the classification loss should encourage the discriminative ability on both action and background.

$$\begin{aligned} \mathcal{L}_{\text{CLS}} &= \mathcal{L}_a + \alpha \mathcal{L}_b \\ &= L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}^a), \mathbf{y}) + \alpha L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}^b), \mathbf{y}^b). \end{aligned} \quad (3)$$

where $L_{ce}(\mathbf{p}, \mathbf{y}) = -\mathbf{y}^\top \log(\mathbf{p})$ is the cross-entropy loss, $p_{cls}(\cdot)$ is the probability output of the action recognition module, and $\mathbf{y}^b \in [0, \dots, 0, 1]$ is the background label.

For action localization, the temporal class activation maps (TCAM) are utilized to locate the key frames that trigger the video-level label. Given a video and the video-level label \mathbf{y} , the TCAM is computed as:

$$\hat{\lambda}_t^a = G(\sigma_s) * \frac{\sum_{c=1}^{C+1} \exp(\mathbf{w}_c \mathbf{x}_t)}{\sum_{c=1}^{C+1} \exp(\mathbf{w}_c \mathbf{x}_t)}. \quad (4)$$

$$\hat{\lambda}_t^b = G(\sigma_s) * \frac{\sum_{c=1}^C \exp(\mathbf{w}_c \mathbf{x}_t)}{\sum_{c=1}^{C+1} \exp(\mathbf{w}_c \mathbf{x}_t)}. \quad (5)$$

where \mathbf{w}_c denotes the parameters of the classification module for class c . $G(\sigma_s)$ is a Gaussian smooth filter with standard deviation σ_s , and $*$ represents convolution. The TCAM $\hat{\lambda}_t^a$ and $\hat{\lambda}_t^b$ are expected to be consistent with the attention $\hat{\lambda}_t$, and thus used to refine the attention via the self-guided loss.

$$\mathcal{L}_{\text{guide}} = \frac{1}{T} \sum_{t=1}^{T+1} |\lambda_t - \hat{\lambda}_t^a| + |\lambda_t - \hat{\lambda}_t^b|. \quad (6)$$

Based on the attention λ , we can preliminarily separate action and background segments in an untrimmed video. Note that the separation is coarse-grained compared with the finer-grained action localization. However, the action and background can provide new perspectives for robust representation learning. As illustrated in the right part of Figure 2, a video \mathcal{V} can be segmented into m action sub-videos $\mathcal{A} = \{v_{[s_i, e_i]}^a\}_{i=1}^m$ and $(m+1)$ background sub-videos $\mathcal{B} = \{v_{[e_{i-1}, s_i]}^b\}_{i=1}^{m+1}$. Specifically, $e_0 = 1$ and $s_{m+1} = T$ are the first and last frame of the video respectively.

B. Intra-Action Shuffling

As discussed earlier, the frame order within an action segment is crucial to understand the semantics. In order to improve the representative ability of the attention module, we perform intra-action shuffling and develop a self-supervised order restoring task. To be specific, we sample non-overlapping clips from a preliminarily segmented action sub-video, and shuffle them to a random order. Using the original order as self-supervision, we learn a clip order prediction model based on the clips' attention-weighted features.

Formally, from each action sub-video $v_{[s_i, e_i]}^a \in \mathcal{A}$, we uniformly sample N fixed-sized clips with intervals in between, denoted as $\{v_{[s_i, k, e_i, k]}^a\}_{k=1}^N \in v_{[s_i, e_i]}^a$ where $s_i \leq s_i, k < e_i, k \leq e_i$. In this way, the scattered clips can describe different phases of the action, and meanwhile they share less resemblance with each other. According to Eq. (1), the feature vector of each clip is $\mathbf{x}_{[s_i, k, e_i, k]}^a$. The sampled N clips are randomly shuffled and organized into a tuple to form the input data, with their original order serving as the target. We formulate order prediction as a classification task, which outputs the probability estimation of the input clip features over different orders. The order prediction module is implemented with the multi-layer perceptron (MLP) structure. The clip features are firstly pairwise concatenated. Each concatenated pair is fed into the ReLU function to obtain a relation vector, i.e., \mathbf{r}_{kj} ,

which captures the relation between the two clips. The relation vectors are further concatenated and go through a FC layer with softmax to arrive at the predicted order \mathbf{p}_{ord} . The order prediction operations are formulated as follows.

$$\mathbf{r}_{kj} = \text{ReLU}(\mathbf{W}_1(\mathbf{x}_{[s_i, k, e_i, k]}^a \| \mathbf{x}_{[s_i, j, e_i, j]}^a) + \mathbf{b}_1). \quad (7)$$

$$\mathbf{p}_{\text{ord}} = \text{softmax}(\mathbf{W}_2(\|\mathbf{r}_{kj}\|) + \mathbf{b}_2). \quad (8)$$

where $\|$ is the concatenation operation of vectors, $k \prec j$ means clip k is in front of clip j , and \mathbf{W}_1 , \mathbf{b}_1 , \mathbf{W}_2 , and \mathbf{b}_2 are parameters of linear transformations.

The order prediction module is optimized with the ordering loss based on cross-entropy function as follows.

$$\mathcal{L}_{\text{intra}} = L_{ce}(\mathbf{p}_{\text{ord}}, \mathbf{y}^{\text{ord}}). \quad (9)$$

where $\mathbf{y}^{\text{ord}} \in \mathbb{R}^{N!}$ is the original order. As we can see, the number of all possible orders, i.e., $N!$, overgrows with the increase in the number of clips. For the sake of efficiency, we set $N = 5$, which yields $5! = 120$ orders.

Intra-action shuffling develops the self-supervised order prediction task, which implicitly benefits representation learning. The ordering loss encourages the attention-based video features to grasp the order-sensitive information within actions, so that the dynamical coherence can be well embedded to enhance the representative ability.

C. Inter-Action Shuffling

Different from the sensitivity to intra-action frame order, the action is relatively location-insensitive. In other words, relocating an action as a whole without altering its inner contents will not affect the semantics. Based on the location-insensitive property, we develop the inter-action shuffling strategy to create new training videos. To be specific, we randomly select several action segments from videos of an identical class, and concatenate them into a new video. It is reasonable that the shared class label is still applicable to the new video. Therefore, the extended videos are naturally attached with video-level labels, and can be safely used as additional training data.

Let $\mathcal{T} = \{(\mathcal{V}^{(l)}, \mathbf{y}^{(l)})\}_{l=1}^L$ denotes the training dataset comprised of L videos with the corresponding video-level labels. Given a predicted action instance $v_{[s_i, e_i]}^{a(l)} \in \mathcal{A}^{(l)} \subset \mathcal{V}^{(l)}$ using attention-based segmentation, we inflate its boundary slightly to obtain the outer-boundary action sub-video $v_{[s_i - \Delta, e_i + \Delta]}^{(l)}$, where Δ is the inflation interval. From the videos of a specific action class $\tilde{\mathbf{y}}$, we randomly select the outer-boundary action sub-videos, and generate a new video $\tilde{\mathcal{V}} = \{v_{[s_i - \Delta, e_i + \Delta]}^{(l)} | l \in [1, L], \mathbf{y}^{(l)} = \tilde{\mathbf{y}}, i \in [1, m^{(l)}]\}$ with video-level label $\tilde{\mathbf{y}}$, where $m^{(l)}$ is the number of predicted action instances in $\mathcal{V}^{(l)}$. Note that, for the generated video, there is no need for feature extraction from scratch. The features $\tilde{\mathbf{X}}$ of generated video $\tilde{\mathcal{V}}$ can be conveniently obtained by simply concatenating the corresponding frame feature vectors, based on which the attention-based representations are also learned. In this way, an additional training dataset $\tilde{\mathcal{T}} = \{(\tilde{\mathcal{V}}^{(l)}, \tilde{\mathbf{y}}^{(l)})\}_{l=1}^{\tilde{L}}$ is created, where \tilde{L} is the number of generated videos.

Datasets	Method	Accuracy
THUMOS14	UNets (Wang et al., 2017) [25]	82.2
	W-TALC (Paul et al., 2018) [5]	85.6
	TSRNet (Zhang et al., 2019) [12]	87.1
	PreTrimNet (Zhang et al., 2019) [8]	89.2
ActivityNet1.2	ActShufNet	92.8
	W-TALC (Paul et al., 2018) [5]	93.2
ActivityNet1.3	ActShufNet	93.8
	TSRNet (Zhang et al., 2019) [12]	91.2
	PreTrimNet (Zhang et al., 2019) [8]	93.3
	ActShufNet	93.4

TABLE I: Comparison of action recognition results on THUMOS14 and ActivityNet (1.2 and 1.3).

To correctly identify action and background in the generated video, the inter-action shuffling module is optimized with the classification loss similar to Eq. (3) as follows.

$$\mathcal{L}_{\text{inter}} = L_{\text{ce}}(p_{\text{cls}}(\tilde{\mathbf{x}}^a), \tilde{\mathbf{y}}) + \alpha L_{\text{ce}}(p_{\text{cls}}(\tilde{\mathbf{x}}^b), \tilde{\mathbf{y}}^b). \quad (10)$$

Inter-action shuffling benefits the representation learning in a weakly supervised learning fashion. On one hand, the training dataset can be effectively expanded by generating additional video-level labeled videos, and meanwhile the problem of data imbalance can be naturally solved by adaptive generation according to data distribution. On the other hand, inter-action shuffling introduces more variety in each generated video, and provides a more challenging auxiliary training dataset to shape a more robust attention model.

D. Global-Local Adversarial Training

In each untrimmed video, action and background segments bear resemblance to some extent in visual and motional aspects. Therefore, action and background can be easily confused with each other, which jeopardizes the localization performance. Recent studies indicate that adversarial training is effective to enhance the model’s tolerance to irrelevant noises, by adding small perturbations to the inputs. To further improve the discriminative ability of the classification module, we develop the global-local adversarial training scheme to achieve robust learning performance.

To be specific, global adversarial training focuses on the robustness of video-level prediction, which is formulated as follows by allowing perturbations to the video-level action and background feature vectors in the classification loss in Eq. (3) with video-level labels.

$$\begin{aligned} \mathcal{L}_{\text{global}} &= \max_{\delta^a} L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}^a + \delta^a), \mathbf{y}) \\ &\quad + \alpha \max_{\delta^b} L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}^b + \delta^b), \mathbf{y}^b). \end{aligned} \quad (11)$$

where δ^a and δ^b are perturbations imposed on \mathbf{x}^a and \mathbf{x}^b , respectively. $\max_{\delta} L_{\text{ce}}(\cdot)$ finds the perturbation that maximizes the loss function and is most likely to fool the classifier. Global adversarial training encourages consistence between video-level prediction and supervision under perturbation, and makes the model less affected by irrelevant contents.

Different from the global scheme, local adversarial training focuses on segments instead of the entire video. It aims to optimize the action-background separation by maximizing the

distinction between adjacent action and background. Formally, for action segment $v_{[s_i, e_i]}^a \in \mathcal{V}$, the adjacent background segments are $v_{[e_{i-1}, s_i]}^b$ and $v_{[e_i, s_{i+1}]}^b$. Local adversarial training encourages different predictions of each adjacent action-background pair, which is formulated as follows.

$$\begin{aligned} \mathcal{L}_{\text{local}} &= \sum_{i=1}^m \max_{\delta^a} \\ &\quad - L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}_{[e_{i-1}, s_i]}^a + \delta_{[e_{i-1}, s_i]}^b), p_{\text{cls}}(\mathbf{x}_{[s_i, e_i]}^a + \delta_{[s_i, e_i]}^a)) \\ &\quad - L_{\text{ce}}(p_{\text{cls}}(\mathbf{x}_{[s_i, e_i]}^a + \delta_{[s_i, e_i]}^a), p_{\text{cls}}(\mathbf{x}_{[e_i, s_{i+1}]}^a + \delta_{[e_i, s_{i+1}]}^b)). \end{aligned} \quad (12)$$

Note that, for all the segments, we use the action representation in Eq. (1) for unified comparison.

Directly calculating δ that maximizes the loss function is infeasible. In this paper, we follow the fast gradient sign method (FGSM) to obtain the perturbation as follows.

$$\delta = \epsilon \text{sign}(\nabla_{\mathbf{x}} f(\mathbf{x})). \quad (13)$$

where ϵ is a pre-defined hyper-parameter, and $f(\mathbf{x})$ is the loss function w.r.t. \mathbf{x} , i.e., $f(\cdot) = L_{\text{ce}}(\cdot)$ for Eq. (11) and $f(\cdot) = -L_{\text{ce}}(\cdot)$ for Eq. (12).

The global-local adversarial training loss can be defined as:

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{global}} + \beta \mathcal{L}_{\text{local}}. \quad (14)$$

Finally, we arrive at the overall loss of ActShufNet.

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \eta \mathcal{L}_{\text{intra}} + \theta \mathcal{L}_{\text{inter}} + \gamma \mathcal{L}_{\text{guide}}. \quad (15)$$

In the training stage, ActShufNet is optimized by minimizing the overall loss in Eq. (15). In the testing stage, the well-trained ActShufNet makes frame-level class activation prediction to achieve temporal localization.

IV. EXPERIMENTS

A. Datasets and Evaluation Metrics

We evaluate the proposed ActShufNet on three benchmarks, i.e. THUMOS14 [43] and two released versions of ActivityNet [44]. As a weakly supervised method, ActShufNet only has access to the video-level annotations during training.

THUMOS14 contains a validation set and a testing set of 1,010 and 1,574 videos, respectively. There are 101 action classes, among which 20 classes are temporally annotated. We focus on the 20-class subset, using the validation set of 200 videos for training and the testing set of 213 videos for evaluation. THUMOS14 is challenging in that it contains videos with multiple actions.

ActivityNet has two released versions, i.e., ActivityNet1.2 and ActivityNet1.3. ActivityNet1.2 contains 100 classes of videos, with 4,819 videos for training, 2,383 for validation, and 2,480 for testing. ActivityNet1.3 is an extension of ActivityNet1.2, which is comprised of 200 activity classes, with 10,024 videos for training, 4,926 for validation, and 5,044 for testing. Since the ground-truth labels for the original testing set are withheld, we adopt the training set for model training and the validation set for testing.

Method	mAP@IoU (%)								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
SCNN (Shou et al., 2016) [24]	47.7	43.5	36.3	28.7	19.0	10.3	5.3	-	-
SSN (Zhao et al., 2017) [1]	66.0	59.4	51.9	41.0	29.8	-	-	-	-
TAL-Net (chao et al., 2018) [2]	59.8	57.1	53.2	48.5	42.8	33.8	20.8	-	-
BSN (Lin et al., 2018) [3]	-	-	53.5	45.0	36.9	28.4	20.0	-	-
P-GCN (Zeng et al., 2019) [4]	69.5	67.8	63.6	57.8	49.1	-	-	-	-
SF-Net (Ma et al., 2020) [37]	68.7	-	54.5	-	34.4	-	16.7	-	-
UNets (Wang et al., 2017) [25]	44.4	37.7	28.2	21.1	13.7	-	-	-	-
STPN (Nguyen et al., 2018) [28]	45.3	38.8	31.1	23.5	16.2	9.8	5.1	2.0	0.3
AutoLoc (Shou et al., 2018) [27]	-	-	35.8	29.0	21.2	13.4	5.8	-	-
W-TALC (Paul et al., 2018) [5]	49.0	42.8	32.0	26.0	18.8	-	6.2	-	-
CMCS (Liu et al., 2019) [6]	53.5	46.8	37.5	29.1	19.9	12.3	6.0	-	-
BaS-Net (Lee et al., 2020) [10]	56.2	50.3	42.8	34.7	25.1	17.1	9.3	3.7	0.5
ActShufNet (UNT)	59.12	53.68	45.25	36.90	27.24	19.47	10.07	4.01	0.39
STPN (Nguyen et al., 2018) [28]	52.0	44.7	35.5	25.8	16.9	9.9	4.3	1.2	0.1
W-TALC (Paul et al., 2018) [5]	55.2	49.6	40.1	31.1	22.8	-	7.6	-	-
TSRNet (Zhang et al., 2019) [12]	55.9	46.9	38.3	28.1	18.6	11.0	5.59	2.19	0.29
TSM (Yu et al., 2019) [38]	-	-	39.5	31.9	24.5	13.8	7.1	-	-
CMCS (Liu et al., 2019) [6]	57.4	50.8	41.2	32.1	23.1	15.0	7.0	-	-
BM (Nguyen et al., 2019) [7]	60.4	56.0	46.6	37.5	26.8	17.6	9.0	3.3	0.4
3C-Net (Narayan et al., 2019) [26]	59.1	53.5	44.2	34.1	26.6	-	8.1	-	-
MAAN (Yuan et al., 2019) [39]	59.8	50.8	41.1	30.6	20.3	12.0	6.9	2.6	0.2
PreTrimNet (Zhang et al., 2019) [8]	57.49	50.73	41.40	32.05	23.09	14.16	7.69	2.33	0.39
DGAM (Shi et al., 2020) [9]	60.0	54.2	46.8	38.2	28.8	19.8	11.4	3.6	0.4
ActionBytes (Jain et al., 2020) [13]	-	-	43.0	35.8	29.0	-	9.5	-	-
Deep Metric Learning (Islam et al., 2020) [40]	62.3	-	46.8	-	29.6	-	9.7	-	-
BaS-Net (Lee et al., 2020) [10]	58.2	52.3	44.6	36.0	27.0	18.6	10.4	3.9	0.5
TSCN (Zhai et al., 2020) [11]	63.4	57.6	47.8	37.7	28.7	19.4	10.2	3.9	0.7
A2CL-PT (Min et al., 2020) [41]	61.2	56.1	48.1	39.0	30.1	19.2	10.6	4.8	1.0
ActShufNet (I3D)	63.44	57.92	48.46	40.01	31.12	22.01	11.26	4.46	0.50

TABLE II: Comparison of action localization results on THUMOS14. Entries are separated regarding the level of supervision. The partition upon the first double horizontal line represents the fully supervised methods. For weakly-supervised setting, we compare both UntrimmedNet (UNT) features and I3D features, as depicted in the lower two partitions.

Method	mAP@IoU (%)			
	0.5	0.75	0.95	Average
SSN (Zhao et al., 2017) [1]	41.3	27.0	6.1	26.6
W-TALC (Paul et al., 2018) [5]	37.0	12.7	1.5	18.0
TSM (Yu et al., 2019) [38]	30.3	19.0	4.5	-
CMCS (Liu et al., 2019) [6]	36.8	22.0	5.6	22.4
3C-Net (Narayan et al., 2019) [26]	35.4	-	-	21.1
CleanNet (Liu et al., 2019) [42]	34.5	22.5	4.9	22.2
DGAM (Shi et al., 2020) [9]	41.0	23.5	5.3	24.4
TSCN (Zhai et al., 2020) [11]	37.6	23.7	5.7	23.6
ActShufNet	41.2	24.9	5.9	25.0

TABLE III: Comparison of action localization results on ActivityNet1.2. The partition upon the double horizontal line represents fully supervised methods, belows are weakly supervised methods. The column Average means the average mAP at IoU thresholds 0.5:0.05:0.95.

Evaluation Metrics. We follow the standard evaluation protocol and report mean Average Precision (mAP) at different intersection over union (IoU) thresholds. The results are calculated using the benchmark code provided by ActivityNet official codebase¹.

B. Implementation Details

We utilize the two-stream I3D networks pre-trained on Kinetics dataset to extract the traditional two-stream features. For the RGB stream, we perform the center crop of size

¹<https://github.com/activitynet/ActivityNet/tree/master/Evaluation>

Method	mAP@IoU (%)			
	0.5	0.75	0.95	Average
SSN (Zhao et al., 2017) [1]	39.12	23.48	5.49	23.98
BSN (Lin et al., 2018) [3]	52.50	33.53	8.85	33.72
P-GCN (Zeng et al., 2019) [4]	42.90	28.14	2.47	31.11
STPN (Nguyen et al., 2018) [28]	29.3	16.9	2.6	-
TSRNet (Zhang et al., 2019) [12]	33.1	18.7	3.32	21.78
TSM (Yu et al., 2019) [38]	30.3	19.0	4.5	-
CMCS (Liu et al., 2019) [6]	34.0	20.9	5.7	21.2
BM (Nguyen et al., 2019) [7]	36.4	19.2	2.9	-
MAAN (Yuan et al., 2019) [39]	33.7	21.9	5.5	-
PreTrimNet (Zhang et al., 2019) [8]	34.8	20.9	5.3	22.5
BaS-Net (Lee et al., 2020) [10]	34.5	22.5	4.9	22.2
A2CL-PT (Min et al., 2020) [41]	36.8	22.0	5.2	22.5
TSCN (Zhai et al., 2020) [11]	35.3	21.4	5.3	21.7
ActShufNet	36.3	23.5	5.8	23.6

TABLE IV: Comparison of action localization results on ActivityNet1.3. The partition upon the double horizontal line represents fully supervised methods, belows are weakly supervised methods. The column Average means the average mAP at IoU thresholds 0.5:0.05:0.95.

224 × 224. For the optical flow stream, we apply the TV-L1 optical flow algorithm. The input to the I3D models are stacks of 16 (RGB or flow) frames sampled at 16 frames per second to obtain two 1024-dimension video features. The model parameters are optimized using the mini-batch stochastic gradient descent with Adam optimizer. The learning rate is set to 1e-4 for both RGB and optical flow streams. We also utilize the dropout operations with ratios 0.5 and common

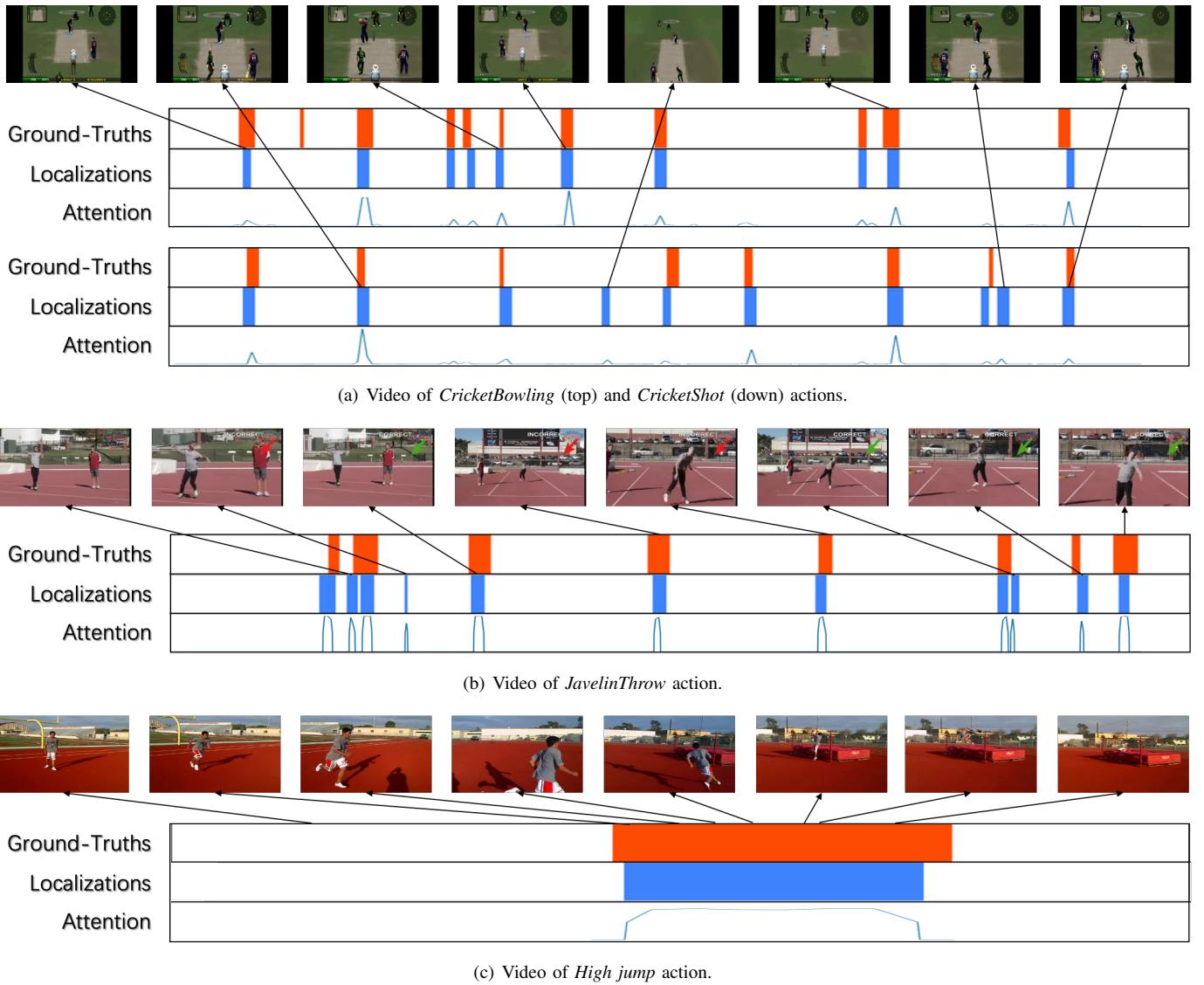


Fig. 3: Qualitative results on THUMOS14 (a and b), and ActivityNet (c). The red bars denote the ground-truth. The blue bars denote localization results.

augmentation techniques including horizontal flipping, cropping augmentation, et al. We set the parameters α , β , ϵ , η , θ and γ to 1, 0.01, 0.001, 1, 0.1 and 0.1, respectively. Our algorithm is implemented in PyTorch.

C. Results

Hyperparameter study. To investigate the effect of training set augmentation via inter-action shuffling, we study the variation of action recognition and localization results on THUMOS14 w.r.t. the number of generated videos. As illustrated in Fig. 4, the performance improves continuously until 2 to 3 times of auxiliary training videos are leveraged, and degrades afterward. Therefore, we expand the training set by 3 times throughout the following experiments.

Action recognition. We compare the action recognition performance of ActShufNet with the state-of-the-art methods. As shown in TABLE I, ActShufNet remarkably outperforms most of its competitors on all three benchmarks. One exception

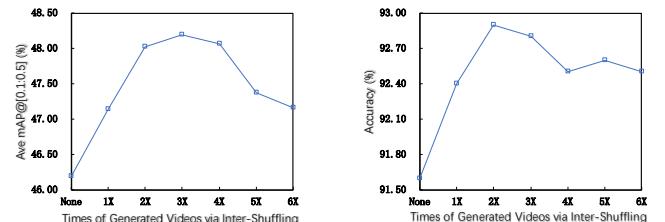


Fig. 4: Action recognition (left) and localization (right) results on THUMOS14 w.r.t. the number of generated videos via inter-shuffling. “*X” means * times of original number of training videos.

is that ActShufNet only slightly surpasses PreTrimNet [8] on ActivityNet1.3. However, it is worth noticing that PreTrimNet is based on fine-grained spatio-temporal segmentation with three-stream features, thus is far more complex than ActShufNet.

Action localization. We evaluate ActShufNet on temporal

\mathcal{L}_{adv}	$\mathcal{L}_{\text{intra}}$	$\mathcal{L}_{\text{inter}}$	mAP@IoU (%)								
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-	-	-	56.22	50.57	40.46	32.05	21.89	12.45	6.81	2.06	0.26
✓	-	-	58.91	53.14	45.98	37.33	27.92	18.99	9.98	3.17	0.35
✓	-	✓	60.42	54.30	46.69	37.93	28.20	19.35	10.80	3.47	0.40
✓	✓	-	60.66	54.63	46.70	38.71	28.91	21.02	11.14	4.26	0.47
✓	✓	✓	63.44	57.92	48.46	40.01	31.12	22.01	11.26	4.46	0.50

TABLE V: Comparison of action localization results of ActShufNet with different implementations on THUMOS14.

action localization task, in comparison with both fully and weakly supervised methods. Localization results on THUMOS14 are listed in TABLE II. The compared methods are compared in chronological order. The lower two partitions are grouped by choice of the feature extractor: UntrimmedNet (UNT) and I3D. It is observed that ActShufNet significantly surpasses its weakly supervised counterparts and achieves the highest average mAP at the same level of supervision, regardless of the feature extractor network. It is especially encouraging to see that ActShufNet even achieves comparable results with some fully supervised methods (in upper parts of the tables).

We also evaluate our ActShufNet on ActivityNet1.2 in TABLE III. We see that our method outperforms all other weakly-supervised approaches. Moreover, despite using generative models (i.e., cVAE), our algorithm outperforms DGAM at all IoU thresholds. Experimental results on ActivityNet1.3 are shown in TABLE IV to compare our method with more methods. Our model outperforms all weakly-supervised methods with the average mAP, following the fully-supervised method with a small gap.

Ablation study. To validate the effectiveness of key components, we compare the full implementation of ActShufNet with its abridged versions without some of the losses in Eq. (15). TABLE V summarizes the action localization results on THUMOS14. We observe that each component is indispensable to achieve accurate results, and the absence of any component will lead to notable performance deterioration. We further illustrate examples of temporal action localization on THUMOS14 ((a) and (b)) and ActivityNet1.3 ((c)), as shown in Fig. 3. As ActivityNet 1.3 is an extension vision of 1.2, we only visualize the results of 1.3 version. The visualization results include (a) a video containing two action classes, (b) short lasting action, and (c) long lasting action. Generally, the multiple actions in a video seems to be similar, which is vulnerable to the boundary noise. In the case (a), the two actions *CricketBowling* and *CricketShot* have overlapping parts along the time axis, our method can also discriminate the actions. In case (b), there are short lasting actions, where actions happen quickly, and our method can catch the key action frames. In case (c), the video contains a complete *High jump* action, which last long time in ground-truths, our method can also track and localize the actions. As we can see, the proposed method is an effective indicator that is capable of locating actions of interest in untrimmed videos under different circumstances.

V. CONCLUSION

In this paper, we have proposed a novel self-augmented framework, namely ActShufNet, for action localization in untrimmed videos with video-level weak supervision. Instead of taking untrimmed video as a whole, we focus on sub-videos derived from preliminary segmentation. Based on analysis of the order-sensitive and location-insensitive properties of actions, we design a two-branch network architecture with intra/inter-action shuffling. The former aims to augment the model’s representative ability via inner-video order shuffling, whereas the latter generates new videos to augment the training set by reorganizing the existing action sub-videos. The global-local adversarial training scheme is further presented to ensure model robustness to irrelevant noises. As demonstrated on three challenging untrimmed video datasets, ActShufNet achieves superior performance over the state-of-the-art weakly supervised methods, and is even comparable to some fully supervised counterparts.

REFERENCES

- [1] 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. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.317>
- [2] 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. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Chao_Rethinking_the_Faster_CVPR_2018_paper.html
- [3] 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. [Online]. Available: https://doi.org/10.1007/978-3-030-01225-0_1
- [4] 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, pp. 7094–7103.
- [5] S. Paul, S. Roy, and A. K. Roy-Chowdhury, “W-TALC: weakly-supervised temporal activity localization and classification,” in *ECCV*, 2018, pp. 588–607. [Online]. Available: https://doi.org/10.1007/978-3-030-01225-0_35
- [6] D. Liu, T. Jiang, and Y. Wang, “Completeness modeling and context separation for weakly supervised temporal action localization,” in *CVPR*, 2019, pp. 1298–1307. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_Completeness_Modeling_and_Context_Separation_for_Weakly_Supervised_Temporal_Action_CVPR_2019_paper.html
- [7] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes, “Weakly-supervised action localization with background modeling,” in *ICCV*, 2019, pp. 5502–5511.
- [8] X. Zhang, H. Shi, C. Li, and P. Li, “Multi-instance multi-label action recognition and localization based on spatio-temporal pre-trimming for untrimmed videos,” in *AAAI*, 2020, pp. 12 886–12 893.
- [9] B. Shi, Q. Dai, Y. Mu, and J. Wang, “Weakly-supervised action localization by generative attention modeling,” in *CVPR*, 2020, pp. 1006–1016.
- [10] P. Lee, Y. Uh, and H. Byun, “Background suppression network for weakly-supervised temporal action localization,” in *AAAI*, 2020, pp. 11 320–11 327. [Online]. Available: <https://aaai.org/ojs/index.php/AAAI/article/view/6793>

- [11] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, and G. Hua, "Two-stream consensus network for weakly-supervised temporal action localization," in *ECCV*, 2020, pp. 37–54.
- [12] X. Zhang, H. Shi, C. Li, K. Zheng, X. Zhu, and L. Duan, "Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision," in *AAAI*, 2019, pp. 9227–9234. [Online]. Available: <https://doi.org/10.1609/aaai.v33i01.33019227>
- [13] M. Jain, A. Ghodrati, and C. G. M. Snoek, "Actionbytes: Learning from trimmed videos to localize actions," in *CVPR*, 2020, pp. 1168–1177.
- [14] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011, pp. 3169–3176. [Online]. Available: <https://doi.org/10.1109/CVPR.2011.5995407>
- [15] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013, pp. 3551–3558. [Online]. Available: <https://doi.org/10.1109/ICCV.2013.441>
- [16] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016, pp. 1933–1941. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.213>
- [17] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015, pp. 4489–4497. [Online]. Available: <https://doi.org/10.1109/ICCV.2015.510>
- [18] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017, pp. 4724–4733.
- [19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456. [Online]. Available: <http://proceedings.mlr.press/v37/ioffe15.html>
- [20] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016, pp. 20–36. [Online]. Available: https://doi.org/10.1007/978-3-319-46484-8_2
- [21] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Two stream LSTM: A deep fusion framework for human action recognition," in *WACV*, 2017, pp. 177–186.
- [22] G. Singh and F. Cuzzolin, "Untrimmed video classification for activity detection: submission to activitynet challenge," 2016. [Online]. Available: <http://arxiv.org/abs/1607.01979>
- [23] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *MM*, 2017, pp. 988–996. [Online]. Available: <https://doi.org/10.1145/3123266.3123343>
- [24] Z. Shou, D. Wang, and S. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016, pp. 1049–1058. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.119>
- [25] L. Wang, Y. Xiong, D. Lin, and L. V. Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017, pp. 6402–6411. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.678>
- [26] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, "3c-net: Category count and center loss for weakly-supervised action localization," in *ICCV*, 2019, pp. 8678–8686.
- [27] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in *ECCV*, 2018, pp. 162–179. [Online]. Available: https://doi.org/10.1007/978-3-030-01270-0_10
- [28] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *CVPR*, 2018, pp. 6752–6761. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Nguyen_Weakly_Supervised_Action_CVPR_2018_paper.html
- [29] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *ICCV*. IEEE Computer Society, 2017, pp. 1338–1347.
- [30] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "Self-supervised spatiotemporal learning via video clip order prediction," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 10334–10343. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Xu_Self-Supervised_Spatiotemporal_Learning_via_Video_Clip_Order_Prediction_CVPR_2019_paper.html
- [31] Z. Luo, B. Peng, D. Huang, A. Alahi, and L. Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 7101–7110. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.751>
- [32] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, "Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 4006–4015. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Self-Supervised_Spatiotemporal_Representation_Learning_for_Videos_by_Predicting_Motion_and_CVPR_2019_paper.html
- [33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.
- [34] T. Miyato, A. M. Dai, and I. J. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=r1X3g2_xl
- [35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: <https://openreview.net/forum?id=rJzIBfZAb>
- [36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.
- [37] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, and Z. Shou, "Sf-net: Single-frame supervision for temporal action localization," in *ECCV*, 2020, pp. 420–437.
- [38] T. Yu, Z. Ren, Y. Li, E. Yan, N. Xu, and J. Yuan, "Temporal structure mining for weakly supervised action detection," in *ICCV*, 2019, pp. 5522–5531.
- [39] Y. Yuan, Y. Lyu, X. Shen, I. W. Tsang, and D. Yeung, "Marginalized average attentional network for weakly-supervised learning," in *ICLR*. OpenReview.net, 2019.
- [40] A. Islam and R. J. Radke, "Weakly supervised temporal action localization using deep metric learning," in *WACV*. IEEE, 2020, pp. 536–545.
- [41] K. Min and J. J. Corso, "Adversarial background-aware loss for weakly-supervised temporal activity localization," in *ECCV*, 2020, pp. 283–299.
- [42] Z. Liu, L. Wang, Q. Zhang, Z. Gao, Z. Niu, N. Zheng, and G. Hua, "Weakly supervised temporal action localization through contrast based evaluation networks," in *ICCV*, 2019.
- [43] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," <http://crcv.ucf.edu/THUMOS14/>, 2014.
- [44] 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. [Online]. Available: <https://doi.org/10.1109/CVPR.2015.7298698>



Xiao-Yu Zhang (Senior Member, IEEE) received the B.S. degree in computer science from Nanjing University of Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

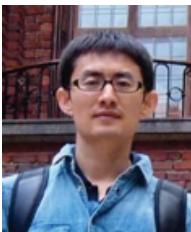
He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He has authored or coauthored more than 60 refereed publications in international journals and conferences. His research interests include artificial intelligence, data mining, computer vision, etc.

Dr. Zhang is a Senior Member of the ACM, CCF, and CSIG. His awards and honors include the Silver Prize of Microsoft Cup of the IEEE China Student Paper Contest in 2009, the Second Prize of Wu Wen-Jun AI Science & Technology Innovation Award in 2016, the CCCV Best Paper Nominate Award in 2017, the Third Prize of BAST Beijing Excellent S&T Paper Award in 2018, and the Second Prize of CSIG Science & Technology Award in 2019.



Haichao Shi received the B.S. degree in software engineering from Beijing Technology and Business University, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree in cyberspace security with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing.

His research interests include pattern recognition, image processing and video content analysis.



Changsheng Li (Member, IEEE) received the B.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013.

He was a Research Assistant with The Hong Kong Polytechnic University, Hong Kong, from 2009 to 2010. He worked with IBM Research-China, Beijing, Alibaba Group, Beijing, and UESTC, respectively. He is currently a Professor with the Beijing Institute of Technology, Beijing. He has authored or coauthored more than 40 refereed publications in international journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEM, the IEEE TRANSACTIONS ON COMPUTERS, PR, CVPR, AAAI, IJCAI, CIKM, MM, ICMR, etc. His research interests include machine learning, data mining, and computer vision.



Xinchu Shi received the B.E. degree from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He was a Research Assistant with Temple University, USA, in 2011 and 2013, and worked as an assistant professor in Institute of Automation from 2014 to 2016. He is currently a senior manager in Meituan Autonomous Delivery (MAD). His research interests include machine learning, computer vision and autonomous driving.