Weakly Supervised Temporal Action Localization through Contrast based Evaluation Networks

Ziyi Liu, *Student Member, IEEE*, Le Wang, *Senior Member, IEEE*, Qilin Zhang, *Member, IEEE*, Wei Tang, *Member, IEEE*, Nanning Zheng, *Fellow, IEEE*, Gang Hua, *Fellow, IEEE*

Abstract—Given only video-level action categorical labels during training, weakly-supervised temporal action localization (WS-TAL) learns to detect action instances and locates their temporal boundaries in untrimmed videos. Compared to its fully supervised counterpart, WS-TAL is more cost-effective in data labeling and thus favorable in practical applications. However, the coarse video-level supervision inevitably incurs ambiguities in action localization, especially in untrimmed videos containing multiple action instances. To overcome this challenge, we observe that significant temporal contrasts among video snippets, *e.g.*, caused by temporal discontinuities and sudden changes, often occur around true action boundaries. This motivates us to introduce a Contrast-based Localization EvaluAtioN Network (CleanNet), whose core is a new temporal action proposal evaluator, which provides fine-grained pseudo supervision by leveraging the temporal contrasts among snippet-level classification predictions. As a result, the uncertainty in locating action instances can be resolved via evaluating their temporal contrast scores. Moreover, the new action localization module is an integral part of CleanNet which enables end-to-end training. This is in contrast to many existing WS-TAL methods where action localization is merely a post-processing step. Besides, we also explore the usage of temporal contrast on temporal action proposal (TAP) generation task, which we believe is the first attempt with the weak supervision setting. Experiments on the THUMOS14, ActivityNet v1.2 and v1.3 datasets validate the efficacy of our method against existing state-of-the-art WS-TAL algorithms.

Index Terms—Action localization, Weakly supervised learning, Temporal Contrast.

1 Introduction

Temporal Action Localization (TAL) aims to detect actions of interest in a video and locate the temporal start and end of each action instance. Thanks to its numerous potential applications such as action retrieval, surveillance, and video summarization [1], [6], [24], [44], TAL has recently drawn increasing attention from the research community. While fully-supervised TAL methods [2], [4], [15], [29], [30], [32], [57], [63] have achieved promising performance, they rely on temporal boundary annotations of all action instances in untrimmed videos. Obtaining this kind of frame-level labels is time-consuming and prohibitively expensive, especially for a large-scale dataset. In this paper, we consider a more cost-effective setting: weakly supervised temporal action localization (WS-TAL), which only requires video-level categorical labels to perform training. It has a great advantage over its fully-supervised counterpart as the video-level labels are much easier to collect.

Currently, many existing WS-TAL methods [31], [34], [38], [47], [53], [61] generate temporal action proposals (TAPs) by directly thresholding the classification score of each video snippet. Achieving localization via such a thresholding pipeline neglects the temporal relations among snippets, which can be critical for action localization. For example, action instances are usually preceded and succeeded by temporal discontinuities, such as special body movements and sudden scene change. As a result, significant "***temporal contrast***" among snippets often occur around true action boundaries. To precisely locate the action boundaries, the potential temporal contrast cue can be leveraged on both WS-TAL task and TAP generation task.

Such observation motivates us to propose a Contrast-based Localization EvaluAtioN Network (CleanNet), which leverages the temporal contrast cue among snippets to generate and evaluate TAPs without temporal annotations. As illustrated in Figure 1, CleanNet consists of four modules respectively designed for feature embedding, action classification, action localization and WS-TAP generation. Given an input untrimmed video, the feature embedding module first extracts snippet-level features. Subsequently, the action classification module produces Snippet-level Classification Predictions (SCPs) and Snippet-level Attention Predictions (SAPs), which are subsequently fused to obtain a video-level prediction. By comparing the video-level predictions and ground truth labels, a classification loss is calculated and the action classification module is trained by minimizing it. Afterwards, the action localization module generates sliding-window-based TAPs and computes the "***contrast score***" of each TAP from their corresponding SCPs and SAPs. The action localization module is trained by maximizing the average contrast score of these survival proposals. After the training of action localization module, the WS-TAP gener-

- *Z. Liu, L. Wang and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: liuziyi@stu.xjtu.edu.cn, {lewang, nnzheng}@mail.xjtu.edu.cn. (Corresponding author: Le Wang.)*
- *Q. Zhang is with ABB Corporate Research Center, Raleigh, NC 27606, USA. E-mail: samqzhang@gmail.com. This research work was carried out before his joining of ABB.*
- *W. Tang is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA. E-mail: tangw@uic.edu.*
- *G. Hua is with Wormpex AI Research, Bellevue, WA 98004, USA. E-mail: ganghua@gmail.com.*

Manuscript received 10 Oct. 2020; revised 23 Mar. 2021; accepted 1 May 2021. Date of publication XX XX 2021; date of current version XX XX 2021. (Corresponding author: Le Wang.)

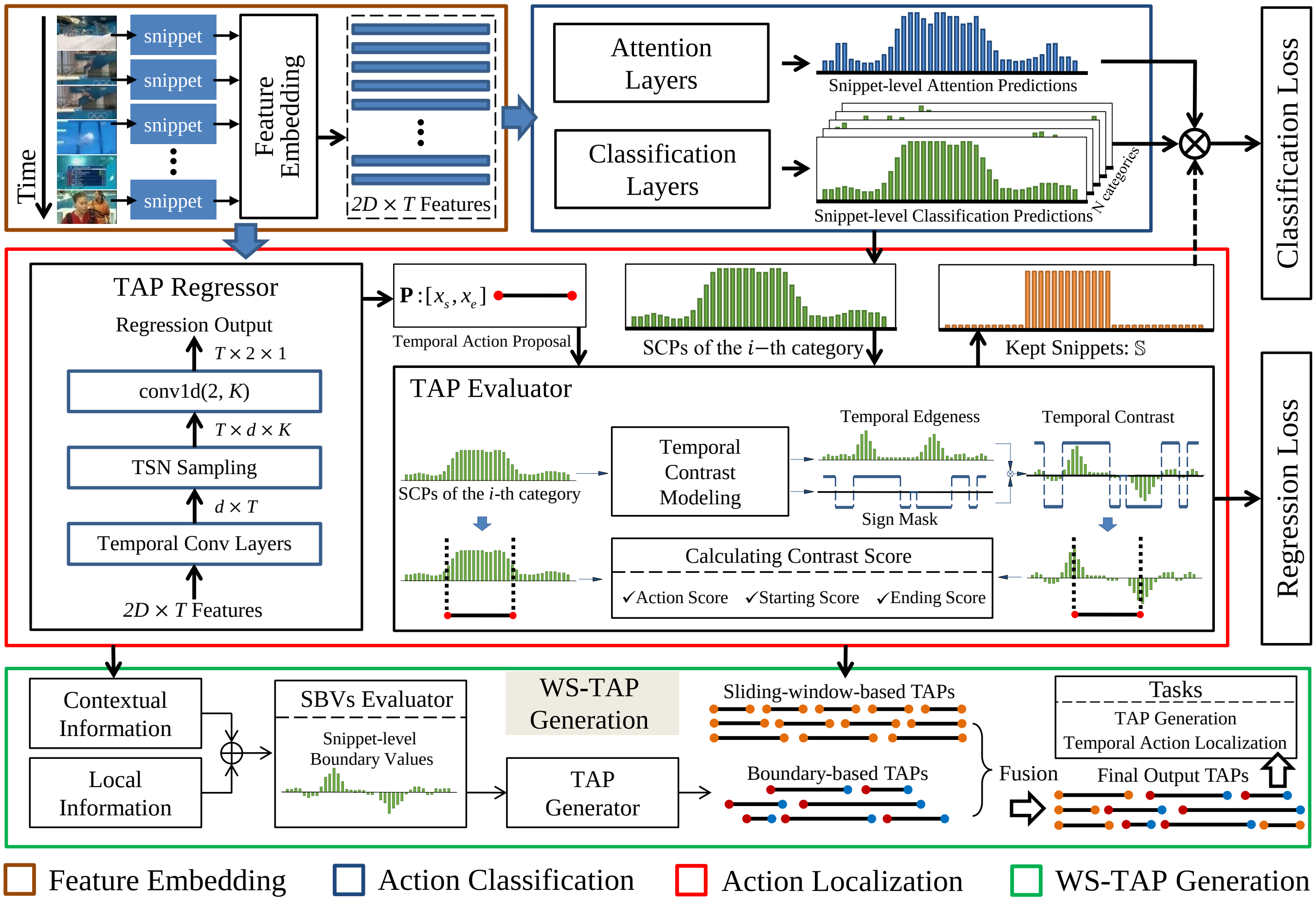


Fig. 1: The proposed CleanNet consists of four components, *i.e.*, a feature embedding module, an action classification module, an action localization module, and a WS-TAP generation module as denoted by brown, blue, red and green rectangles, respectively. Training inputs: untrimmed videos with video-level categorical labels. Prediction outputs: action instance category labels with temporal starts, ends and confidence scores.

ation module generates boundary-based TAPs and fuses them with the sliding-window-based TAPs to obtain the final TAL and TAP generation results.

Specifically, the contrast score of each TAP composes an action score and two edge scores (the starting and ending scores). They represent the likelihood of the TAP containing a specific action and the consistency of the TAP starting/ending with specific action edges, respectively. By combining these scores, the comprehensive contrast score can measure both the content and the completeness of TAPs. Thus, TAPs with higher contrast scores are more likely to be true action instances. Moreover, in CleanNet, there is an interaction between action classification and action localization, and they benefit each other. On one hand, action classification provides SCPs to the TAP evaluator, which serve as the basis to compute the contrast scores of TAPs. On the other hand, action localization offers localization-based filtering of irrelevant frames, as illustrated by the dashed arrow in the upper right corner of Figure 1, where irrelevant snippets are regarded as containing no target action when calculating the classification loss.

Similar to the anchor-based 2D detection methods [40], the TAP regressor in the action localization module adopts multi-scale sliding windows with regression for TAP generation. However, the sliding-window-based TAPs generated by such a pipeline could be inflexible to fully account for the variations of the action instance durations. To address this problem, the WS-TAP generation module generates boundary-based TAPs, based on the temporal contrast modeled by the TAP evaluator. It accommodates flexible action durations and provides more accurate temporal boundaries, as shown in the green box of Figure 1.

Existing TAP generation methods are proposed for fully supervised TAL methods [9], [13], [15], [30], [44], [63]; while in the weakly supervised setting, TAP generation is not a well explored task. Motivated by the fact that regardless of its temporal durations, any TAP must contain a start and an end boundary, we propose to obtain TAPs with flexible durations by connecting potential starts and ends indicated by the temporal contrast cue.

Taking SCPs and temporal contrast as inputs, the WS-TAP generation module first obtains the snippet-level boundary values (SBVs) by accounting for both local and contextual information (as illustrated in Figure 4). With SBVs, starting and ending proposals (collectively referred to as boundary proposals) are generated and evaluated. Subsequently, boundary-based TAPs are obtained by exhaustively connecting one starting proposal with one ending proposal (as illustrated in Figure 5). Afterwards, the boundary-based TAPs and sliding-window-based TAPs are fused by a TAP fusion scheme to obtain the final TAPs. Finally, these TAPs together with their contrast scores calculated by the TAP evaluator are collected as the results for WS-TAL and TAP generation tasks.

In summary, the key contributions of this paper include (1) a new TAP evaluator that quantifies the temporal contrast among SCPs to facilitate WS-TAL; (2) a new WS-TAP generation pipeline with a concept of boundary proposals, which we believe is the first one readily applicable to weakly supervised setting. (3) a CleanNet with a trainable action localization module for WS-TAL, where action classification and localization are mutually beneficial; (4) the state-of-the-art WS-TAL performance on three benchmarks, which are even comparable to some fully-supervised TAL methods.

This paper extends our conference paper [64] in four respects. First of all, this paper explores the TAP generation task in the weakly supervised setting, by extending CleanNet with a new WS-TAP generation module, summarized as in contribution (2). Corresponding experiments on the WS-TAP generation module validate its contribution on both TAP generation and WS-TAL tasks. Second, more comparisons and additional qualitative results are presented to validate the superiority of our CleanNet over thresholding-based methods. Moreover, more discussions including strengths and limitations of our method are provided along with additional qualitative results. Last but not least, more extensive experiments are carried out to compare the proposed method and its ablated variants.

The rest of the paper is organized as follows. Section 2 discusses related work about action recognition, fully and weakly supervised TAL. Afterwards, we present the technical details of our proposed CleanNet in Section 3. Section 4 introduces how to train the proposed CleanNet. Experimental results and discussions are presented in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

We briefly review related work in action recognition, TAP generation with full supervision, TAL with full supervision and weak supervision.

2.1 Action Recognition

Prior to the prevalence of deep neural networks, action recognition models rely on hand-crafted features [8], [25], [41], [51]. Among them, the improved Dense Trajectory (iDT) [51] achieves the best performance. Recently, Convolutional Neural Networks (CNNs) have emerged as the state-of-the-art visual feature extractor and numerous CNN-based action recognition methods are proposed. Two-stream networks [12], [45] take as input optical flow in addition to images in a two-stream architecture, and a fusion of spatial and temporal features is carried out to obtain action recognition results. 3D convolutional networks (3D ConvNets) [22], [49], [50] take video clips as input to acquire spatial and temporal correlations among video frames. Temporal segment networks (TSN) [54] capture the long-range temporal structure with sparse sampling for action recognition. Inflated 3D ConvNet (I3D) [3] combines two-stream networks with 3D convolutions to further boost the action recognition accuracy. This architecture is also widely used as feature backbone in other tasks [20], [55]. Wu *et al*. [56] propose a long-term feature bank extracted from the whole video to help CNNs model long-term information. Slowfast Networks [11] leverage a fast pathway and a slow pathway to capture motion and spatial semantics respectively, and both of them are exploited for video recognition.

2.2 TAL with Full Supervision

Different from the action recognition task which only requires video-level categorical predictions, TAL predicts not only a categorical label for each action instance but also their respective temporal boundaries (*i.e.*, starts and ends). Fully-supervised TAL methods require both types of annotations during training. Besides, different from the online action detection task [7], [18], the TAL is an offline task, *i.e.*, the localization is done after given the whole video.

Thanks to the advancements of deep learning-based object detection methods, such as R-CNN [17] and its variants [16], [40], many approaches follow a similar pipeline of "generating and then classifying TAPs" to perform TAL [2], [4], [5], [9], [15], [44], [57], [63]. Some of these works [4], [5], [15] further adjust the Faster R-CNN architecture to resolve the receptive field issues and make better use of the contextual information. Recently, the dependencies among temporal action proposals are considered. BMN [29] proposes a boundary-matching mechanism to mine the context information of neighboring proposals. DBG [28] further leverages proposal-level information for boundary generation and regression. P-GCN [62], G-TAD [58] and AGCN [27] capture the proposal-proposal relations leveraging the Graph Convolutional Networks (GCN).

2.3 TAP Generation with Full Supervision

Following popular object detection methods such as the R-CNN and its variants [16], [17], [40], the simplest strategy to generate TAPs is to exhaustively apply multi-scale sliding windows or pre-defined temporal durations [2], [9], [36], [44], [52], [60]. Some other works [4], [5], [15], [57] exploit the Faster R-CNN architecture [40], and they use anchors and boundary regression to generate high-quality proposals.

For non-sliding-window-based methods, TAG [63] adopts the watershed algorithm to generate proposals with flexible durations and more accurate boundaries. CTAP [13] proposes a proposal complementary filter to better fuse the proposals from sliding windows and TAG [63]. BSN [30] introduces a local-to-global procedure to locate and evaluate proposals by combining high-probability boundaries. These special adjustments are reportedly responsible for the improvements in TAP generation and TAL performance over sliding-window-based ones.

2.4 TAL with Weak Supervision

The idea of performing TAL using only video-level categorical annotations (*i.e.*, WS-TAL) was first introduced in [48]. Hide-and-Seek [47] randomly hides regions to encourage the model to focus on both the most discriminative parts and other relevant parts of the target. UntrimmedNet [53] uses a soft selection module to locate target temporal action segments, which is similar to temporal attention weights, and the final localization is achieved by thresholding these segments after the scoring. Nguyen *et al.* [34] propose a sparse loss function to facilitate the selection of segments. Paul *et al.* [38] propose a co-activity loss and combine it with a multiple instance learning loss to train a weakly-supervised network. The localization parts of these methods are all based on thresholding on the final SCPs. Yuan *et al.* [61] propose a marginalized average attentional network that can locate the entire action by suppressing the response of the most salient regions. Recently, many methods focus on "background suppression" to improve the quality of snippet-level classification. Liu *et al.* [31] generate hard negative videos for context separation and model completeness using multi-branches with a diversity loss. Nguyen *et al.* [35] propose background modelling for better background suppression, and other unsupervised losses to guide the attention to achieve better TAL performance. Lee *et al.* [26] introduce an auxiliary class for background, based on which, an asymmetrical training strategy is designed to suppress activations from background frames. Huang *et al.* [19] adopt a clustering loss to separate actions from backgrounds and learn intra-compact features. Min *et al.* [33] focus on using triplets to distinguish background features from activity-related features to achieve action-background separation. Essentially, these methods reformulate TAL into a snippet-by-snippet action classification task. The final action localization is achieved by a simple thresholding over the snippet-level classification results. Despite its simplicity, this reformulation is arguably suboptimal, and it leaves a large room for performance improvement.

The recent AutoLoc [43] directly predicts the temporal boundaries of each action instance by benefiting from its "outer-inner-contrastive loss". The proposed CleanNet is distinctive from AutoLoc in the following three aspects. First of all, our TAP evaluator exploits temporal contrast instead of depending only on the average score diffidence between inner and outer region of proposals, and treats the starting/ending boundaries separately to achieve better robustness to noise. Second, the action classification and action localization in CleanNet are interdependent and mutually beneficial, while the counterparts in AutoLoc are independent. Moreover, our TAP regressor is specially designed to address the receptive field issue in the temporal dimension. All these three differences contribute to the superiority of the proposed CleanNet, as discussed below in Section 5.2.1.

Besides, most aforementioned WS-TAL methods do not involve TAP generation because their TAL results are obtained by thresholding. Only AutoLoc [43] and CleanNet [64] involve TAP generation. They both rely on a "multi-scale sliding windows and regression" manner for TAP generation. However, such a TAP generation pipeline could be inefficient and inflexible to fully account for the variations of the action instance durations. Therefore, we extend our method with a new WS-TAP generation module, which can provide boundary-based TAPs with flexible temporal durations.

3 Proposed CleanNet

In this section, we introduce the proposed Contrast-based Localization EvaluAtioN Network (CleanNet). As illustrated in Figure 1, CleanNet composes four major components, *i.e.*, the feature embedding module, the action classification module, the action localization module and the WS-TAP generation module. The input videos are first processed by the feature embedding module to obtain snippet-level features. Afterwards, the obtained features are passed to the action classification module to produce Snippet-level Classification Predictions (SCPs) and Snippet-level Attention Predictions (SAPs). After acquiring SCPs and SAPs, the action localization module refines the temporal locations of TAPs via a TAP regressor and scores them using the customized TAP evaluator to obtain sliding-window-based TAPs. Afterwards, the WS-TAP generation module generates boundary-based TAPs and fuse them with sliding-window-based TAPs to obtain the final results.

3.1 Snippet-Level Feature Embedding

The input to the feature embedding module (the brown rectangle in Figure 1) is untrimmed videos, and the output is the corresponding snippet-level features. Its design mainly follows UntrimmedNet [53]. After dividing each video into non-overlapping snippets of the same length (*i.e.*, 15 frames), temporal features are extracted snippet-after-snippet, which are referred to as snippet-level features $\mathbf{F}$. The temporal granularity of 15 video frames (approximately 0.5 second) has been empirically demonstrated to be sufficient for TAL.

The feature embedding backbone is the TSN [54] with the Inception network architecture and Batch Normalization [21]. The pre-trained spatial stream (an RGB input) and the temporal stream (an optical flow input) are trained individually. The obtained D-dimensional ($D = 1024$) outputs after the `global_pool` layers from both streams are concatenated as one snippet-level feature. Specifically, for an input video with T snippets ($15T$ video frames), the dimension of the output $\mathbf{F}$ is $2D$ channels by T snippets. The feature of the t-th snippet is denoted as $\mathbf{F}(t) \in \mathbb{R}^{2D\times 1}$.

3.2 Action Classification

With $\mathbf{F} \in \mathbb{R}^{2D\times T}$, the action classification module (the blue rectangle in Figure 1) computes both the snippet-level classification predictions (SCPs) and the snippet-level attention predictions (SAPs) with two groups of fully connected layers, respectively. SCPs and SAPs are respectively denoted as $\mathbf{\Psi} \in \mathbb{R}^{N\times T}$ and $\boldsymbol{\varphi} \in \mathbb{R}^{1\times T}$, where N and T are the number of action categories and the number of snippets, respectively. Since the structure of our action classification module is the same as that of UntrimmedNet [53], a straightforward practice to obtain $\mathbf{\Psi}$ and $\boldsymbol{\varphi}$ is to average the outputs of UntrimmedNet from both spatial and temporal streams. To make this fusion step trainable, we design our action classification module as follows,

$$\mathbf{\Psi}(t) = (\mathbf{\Psi}^r(t) + \mathbf{\Psi}^f(t))/2, \tag{1}$$

$$\begin{bmatrix} \mathbf{\Psi}^r(t) \\ \mathbf{\Psi}^f(t) \end{bmatrix} = \mathbf{W}^c\mathbf{F}(t) + \mathbf{b}^c, \tag{2}$$

$$\boldsymbol{\varphi}(t) = \mathbf{w}^a \cdot \mathbf{F}(t) + b^a, \tag{3}$$

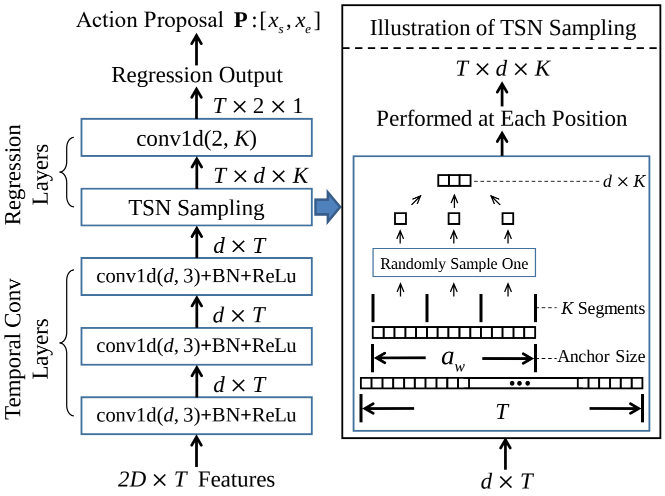


Fig. 2: The structure of the TAP regressor. Input snippet-level features are fed into three stacked temporal convolutional layers before a *TSN sampling* layer, which matches its receptive field size with the anchor size.

where $t = 1, \ldots, T$ is the snippet index. $\mathbf{\Psi}^r(t) \in \mathbb{R}^{N\times 1}$ and $\mathbf{\Psi}^f(t) \in \mathbb{R}^{N\times 1}$ are the action classification predictions of the t-th snippet from the spatial stream and temporal stream, respectively. $\mathbf{W}^c \in \mathbb{R}^{2N\times 2D}$ and $\mathbf{b}^c \in \mathbb{R}^{2N\times 1}$ are the parameters of the classification layer. $\mathbf{w}^a \in \mathbb{R}^{1\times 2D}$ and b^a are the parameters of the attention layer. They are initialized as

$$\mathbf{W}^c = \begin{bmatrix} \mathbf{W}^{c_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}^{c_f} \end{bmatrix}, \ \mathbf{b}^c = \begin{bmatrix} \mathbf{b}^{c_r} \\ \mathbf{b}^{c_f} \end{bmatrix}, \tag{4}$$

$$\mathbf{w}^a = \frac{1}{2}\begin{bmatrix} \mathbf{w}^{a_r} & \mathbf{w}^{a_f} \end{bmatrix}, b^a = \frac{b^{a_r} + b^{a_f}}{2}, \tag{5}$$

where $\mathbf{W}^{c_r} \in \mathbb{R}^{N\times D}$, $\mathbf{W}^{c_f} \in \mathbb{R}^{N\times D}$, $\mathbf{w}^{a_r} \in \mathbb{R}^{1\times D}$ and $\mathbf{w}^{a_f} \in \mathbb{R}^{1\times D}$ stand for the weights of the classification and attention layers with an RGB input and an optical flow input, respectively. $\mathbf{b}^{c_r} \in \mathbb{R}^{N\times 1}$, $\mathbf{b}^{c_f} \in \mathbb{R}^{N\times 1}$, b^{a_r} and b^{a_f} are the corresponding bias parameters. They are initialized by loading the pre-trained UntrimmedNet models[1]. After initialization, our action classification module fuses the two-stream outputs from the pre-trained UntrimmedNet and remains trainable for further finetuning. Finally, for each video with T snippets, we obtain its SCPs ($\mathbf{\Psi} \in \mathbb{R}^{N\times T}$) and SAPs ($\boldsymbol{\varphi} \in \mathbb{R}^{1\times T}$).

3.3 Action Localization

The main contribution of this paper lies in the design of the action localization module (the red rectangle in Figure 1). Unlike existing TAL methods shoehorning the action localization component as a post-processing procedure (typically by thresholding), our action localization module is embedded into the proposed CleanNet, and it composes a TAP regressor and a TAP evaluator. Thanks to the new TAP evaluator, TAPs with higher contrast scores are more likely to be true action instances (as validated in Section 5.2.1).

3.3.1 TAP Regressor

The goal of the TAP regressor is to regress TAPs from multi-scale sliding windows to cover the temporal range of each action instance via temporal boundary regression. Inspired by existing anchor-based 2D bounding box regression techniques [39], [40], we utilize similar settings in this 1D temporal regression. Specifically, for an anchor with a temporal duration (*i.e.*, the number of snippets) a_w and a temporal location τ, its boundary regression target is two values: r_c is relevant to the regressed center and r_w is relevant to the regressed duration. As shown in Figure 2, the anchors are generated by dense sampling and the regression target is predicted by regression layers. Let $\mathbf{P}$ denote the regressed anchor (*i.e.*, one TAP). Its centroid x_c and temporal duration x_w are obtained as

$$x_c = a_w \cdot r_c + \tau, \tag{6}$$

$$x_w = a_w \cdot \exp(r_w). \tag{7}$$

The starting and ending boundaries of $\mathbf{P}$ are calculated as

$$x_s = x_c - x_w/2, \tag{8}$$

$$x_e = x_c + x_w/2. \tag{9}$$

For simplicity, we use $[x_s, x_e]$ to parameterize $\mathbf{P}$.

However, such a naive adaptation of the spatial bounding box regression algorithm is insufficient due to some potential receptive field issues. More specifically, the spatial regression results in [40] are obtained from a 1×1 convolution layer upon the output of `pool5` in VGG16 [46], achieving a receptive field of 212, which is large enough given the input image resolution of 224×224. If this strategy is directly applied to 1D temporal regression, the receptive field of snippet-level features ($\mathbf{F} \in \mathbb{R}^{2D\times T}$) along the temporal dimension is merely 1, since they are extracted snippet-by-snippet. Thus, it is unrealistic to expect reasonable regression outputs when the size of the receptive field is much smaller than that of the anchor.

A straightforward remedy might be stacking multiple temporal convolutional layers upon snippet-level features $\mathbf{F}$, but the increase of the receptive field size is still limited. To match the size of the receptive field with the corresponding anchor size, we exploit a sparse temporal sampling strategy inspired by TSN [54]. In detail, we divide each anchor into K segments and randomly sample a temporal location per segment, and then obtain a fixed size (K) representation regardless of the anchor size. We term this strategy as *TSN sampling*, as illustrated in Figure 2. Subsequently, the sampled features are fed into another convolutional layer to predict the regression values.

3.3.2 TAP Evaluator

To supervise the TAP regressor, we need to evaluate the quality of each regressed TAP. In the fully-supervised TAL setting where manually labeled temporal boundaries are available, TAPs can be readily evaluated by comparing them with ground truth using a metric such as Intersection-over-Union (IoU). However, in the WS-TAL setting where explicit temporal boundary annotations are unavailable, the design of the TAP evaluator is nontrivial.

In CleanNet, we propose a new TAP evaluator to provide pseudo-supervision based on SCPs of the entire video. The intuition of exploiting all SCPs is to promote complete TAPs with correct contents while penalizing fragmented short ones. The workflow of the TAP evaluator is illustrated in

[1] https://github.com/wanglimin/UntrimmedNet

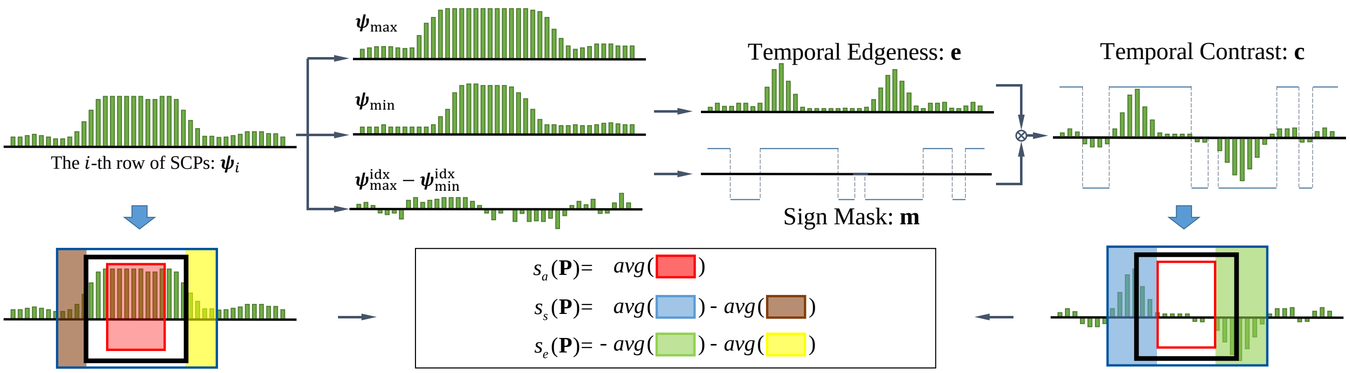

The i-th row of SCPs: $\boldsymbol{\psi}_i$
$\boldsymbol{\psi}_{\max}$
$\boldsymbol{\psi}_{\min}$
$\boldsymbol{\psi}^{\text{idx}}_{\max} - \boldsymbol{\psi}^{\text{idx}}_{\min}$
Temporal Edgeness: $\mathbf{e}$
Sign Mask: $\mathbf{m}$
$\otimes$
Temporal Contrast: $\mathbf{c}$
$s_a(\mathbf{P})$= avg(🟥)
$s_s(\mathbf{P})$= avg(🟦) - avg(🟫)
$s_e(\mathbf{P})$= - avg(🟩) - avg(🟨)

Fig. 3: The workflow of the TAP evaluator in CleanNet. To locate action instances of the i-th category in a video, the inputs to the evaluator are all temporal SCPs corresponding to the i-th action category $\boldsymbol{\psi}_i \in \mathbb{R}^{1\times T}$ (illustrated as the green histogram) and an arbitrary TAP $\mathbf{P}$ (denoted as the black bounding boxes imposed on the green histogram). The output is the contrast score $s(\mathbf{P})$ of $\mathbf{P}$, according to Eq. (17).

Figure 3. To locate action instances of the i-th category ($i = 1, \ldots, N$) in a video, the input to the evaluator is all temporal SCPs corresponding to the i-th action category, *i.e.*, $\boldsymbol{\psi}_i \in \mathbb{R}^{1\times T}$ (the i-th row of $\boldsymbol{\Psi}$, illustrated as a green histogram, provided by the action classification module) and an arbitrary TAP $\mathbf{P}$ (illustrated as the bolded black bounding boxes imposed on the histograms on bottom corners). To simplify the subscripts of subsequent $\boldsymbol{\psi}$ variants, we temporarily drop the subscript of $\boldsymbol{\psi}_i$ as $\boldsymbol{\psi}$ in Section 3.3.2.

To account for the temporal contrast information, we propose the temporal edgeness vector $\mathbf{e} \in \mathbb{R}^{1\times T}$ as

$$\mathbf{e} = (\boldsymbol{\psi}_{\max} - \boldsymbol{\psi}_{\min}) \odot [\text{abs}(\boldsymbol{\psi}^{\text{idx}}_{\max} - \boldsymbol{\psi}^{\text{idx}}_{\min})]^{-1}, \tag{10}$$

where $\odot$ indicates element-wise multiplication, abs$(\cdot)$ and $[\cdot]^{-1}$ represent the element-wise absolute value and reciprocal function, respectively. $\boldsymbol{\psi}_{\max} \in \mathbb{R}^{1\times T}$ is derived by sliding a max pooling window[2] upon $\boldsymbol{\psi}$, and $\boldsymbol{\psi}^{\text{idx}}_{\max} \in \mathbb{R}^{1\times T}$ is the corresponding index vector of local maximums. Similarly, $\boldsymbol{\psi}_{\min} \in \mathbb{R}^{1\times T}$ and $\boldsymbol{\psi}^{\text{idx}}_{\min} \in \mathbb{R}^{1\times T}$ are the min pooling values and indexes, respectively. Intuitively, the temporal edgeness $\mathbf{e}$ represents the likelihood of each snippet being the boundary of an action instance. To distinguish the starts and ends of action instances (*i.e.*, the rising and falling edges in $\boldsymbol{\psi}$), a sign mask $\mathbf{m} \in \mathbb{R}^{1\times T}$ is defined as

$$\mathbf{m}(t) = \begin{cases} 1 & \text{if } \boldsymbol{\psi}^{\text{idx}}_{\min}(t) \leqslant t < \boldsymbol{\psi}^{\text{idx}}_{\max}(t), \\ -1 & \text{if } \boldsymbol{\psi}^{\text{idx}}_{\min}(t) > t \geqslant \boldsymbol{\psi}^{\text{idx}}_{\max}(t), \\ 0 & \text{otherwise}, t = 1, \ldots, T. \end{cases} \tag{11}$$

Subsequently, the temporal contrast $\mathbf{c} \in \mathbb{R}^{1\times T}$ is obtained as

$$\mathbf{c} = \mathbf{m} \odot \mathbf{e}, \tag{12}$$

which is illustrated as the histogram on the top-right in Figure 3. Intuitively, the absolute value of $\mathbf{c}$ represents the likelihood of each snippet being the boundary of an action instance. Positive and negative values indicate the starting and ending boundaries of action instances, respectively.

For an arbitrary TAP $\mathbf{P}$:$[x_s, x_e]$, we compute its inflated and deflated regions $\mathbf{P}^{\text{inf}}$:$[x^{\text{inf}}_s, x^{\text{inf}}_e]$, $\mathbf{P}^{\text{def}}$:$[x^{\text{def}}_s, x^{\text{def}}_e]$ as

$$\begin{array}{ll} x^{\text{inf}}_s = x_s - x_w/4, & x^{\text{inf}}_e = x_e + x_w/4, \\ x^{\text{def}}_s = x_s + x_w/4, & x^{\text{def}}_e = x_e - x_w/4, \end{array} \tag{13}$$

which are illustrated as the blue and red bounding boxes imposed on the histograms on bottom corners in Figure 3, respectively. Definitions of x_c and x_w are included in Section 3.3.1.

With $\boldsymbol{\psi}$, $\mathbf{c}$, $\mathbf{P}$, $\mathbf{P}^{\text{inf}}$ and $\mathbf{P}^{\text{def}}$, three scores are calculated, *i.e.*, the action score $s_a(\mathbf{P})$ represents the likelihood of $\mathbf{P}$ containing a specific action instance, the starting score $s_s(\mathbf{P})$ reflects the likelihood of $\mathbf{P}$'s start stage coinciding with the beginning of an action instance, and the ending score $s_e(\mathbf{P})$ indicates the likelihood of $\mathbf{P}$'s end stage coinciding with the ending of an action instance. They are

$$s_a(\mathbf{P}) = \text{avg}(\boldsymbol{\psi}(x^{\text{def}}_s : x^{\text{def}}_e)), \tag{14}$$

$$s_s(\mathbf{P}) = \text{avg}(\mathbf{c}(x^{\text{inf}}_s : x^{\text{def}}_s)) - \text{avg}(\boldsymbol{\psi}(x^{\text{inf}}_s : x_s)), \tag{15}$$

$$s_e(\mathbf{P}) = -\text{avg}(\mathbf{c}(x^{\text{def}}_e : x^{\text{inf}}_e)) - \text{avg}(\boldsymbol{\psi}(x_e : x^{\text{inf}}_e)), \tag{16}$$

where avg$(\cdot)$ denotes arithmetic average. Intuitively, the first term in Eq.(15)/Eq.(16) indicates the average contrast around the starting/ending phase of $\mathbf{P}$. The second term in Eq.(15)/Eq.(16) indicates the actionness before/after the starting/ending boundary of $\mathbf{P}$. A good starting/ending boundary should satisfy that the average contrast around it is high/low (*i.e.*, the first term in Eq.(15)/Eq.(16)), while the actionness before/after it is low (*i.e.*, the second term in Eq.(15)/Eq.(16)). Therefore, the starting and ending scores of $\mathbf{P}$ are designed as Eq.(15) and Eq.(16), respectively.

The final contrast score $s(\mathbf{P})$ is a weighted summation,

$$s(\mathbf{P}) = s_a(\mathbf{P}) + \frac{1}{2}\left(s_s(\mathbf{P}) + s_e(\mathbf{P})\right). \tag{17}$$

By summing up action scores and edge scores, the contrast score accounts for both the content and context of $\mathbf{P}$, which promotes completeness and continuity in TAP evaluation. We adopt such a scoring scheme because an ideal TAP should pinpoint the starting and ending temporal boundaries of an action instance. If $\mathbf{P}$ has a high contrast score, it indicates that $\mathbf{P}$ has correct content, *i.e.*, avg$(\boldsymbol{\psi}(x^{\text{def}}_s : x^{\text{def}}_e))$ is high, as well as it coincides with the start and end stages of an action instance. For example, our contrast score can avoid fragmented short TAPs with only correct contents but

[2] The max pooling kernel size is 7. To ensure the output $\boldsymbol{\psi}_{\max}$ is identical in size with the input $\boldsymbol{\psi}$, the stride and padding are 1 and 3, respectively.

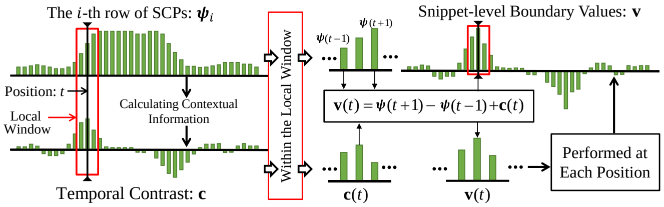

The i-th row of SCPs: $\boldsymbol{\psi}_i$
Position: t
Local Window
Calculating Contextual Information
Temporal Contrast: $\mathbf{c}$
Within the Local Window
$\boldsymbol{\psi}(t-1)$
$\boldsymbol{\psi}(t+1)$
Snippet-level Boundary Values: $\mathbf{v}$
$\mathbf{v}(t)=\boldsymbol{\psi}(t+1)-\boldsymbol{\psi}(t-1)+\mathbf{c}(t)$
$\mathbf{c}(t)$
$\mathbf{v}(t)$
Performed at Each Position

Fig. 4: Workflow of the SBVs evaluator. The input is the SCPs of the n-th action illustrated as a histogram in the upper left corner, denoted as $\boldsymbol{\psi}_c \in \mathbb{R}^{1\times T}$. The output SBVs ($\mathbf{b} \in \mathbb{R}^{1\times T}$) is illustrated as the histogram in the upper right corner. At each temporal position t, $\mathbf{b}(t)$ is calculated based on data within the local window (illustrated as the red bounding box superimposed on the histograms). As the local window slides through all T temporal positions, SBVs of shape $1 \times T$ (T being the number of snippets) are computed.

poor boundaries. Moreover, Section 5.2.1 also validates the contributions of each term in Eq. (17) and all terms in it, *i.e.*, $s_a(\mathbf{P})$, $s_s(\mathbf{P})$, and $s_e(\mathbf{P})$, are indispensable components of the proposed contrast score $s(\mathbf{P})$.

3.4 WS-TAP Generation

The main extension of our method compared with its conference version [64] lies in the WS-TAP generation module. Similar to the anchor-based 2D detection methods [40], the TAP regressor in the action localization module adopts multi-scale sliding windows with regression for TAP generation. However, sliding-window based TAPs generated from such a pipeline could be inflexible to fully account for the variations of the action instance durations. To solve this problem, we proposed a WS-TAP generation module based on the temporal contrast $\mathbf{c}$ in the TAP evaluator. It accommodates flexible action durations and provides more accurate temporal boundaries by generating boundary-based TAPs.

The boundary-based TAPs are generated in two steps. First, snippet-level boundary values (SBVs) are derived as illustrated in Figure 4. Second, potential starting and ending boundaries are formulated as boundary proposals and subsequently selected and connected to generate boundary-based TAPs, as illustrated in Figure 5. To differentiate sliding-window-based TAPs set (obtained by TAP regressor in Section 3.3.1) and the boundary-based TAPs set, we term them as $\mathbb{P}^w = \{\mathbf{P}^w\}$ and $\mathbb{P}^b = \{\mathbf{P}^b\}$, respectively.

3.4.1 SBVs Evaluator

The objective of the SBVs evaluator is to reveal the likelihood of each snippet being a starting or ending boundary. Based on these likelihood values, potential starting and ending boundaries are subsequently located.

The workflow of the SBVs evaluator is illustrated in Figure 4. When locating the i-th action category ($i = 1, \dots, N$) in an untrimmed video, the input of the SBVs evaluator is SCPs corresponding to that action, *i.e.*, $\boldsymbol{\psi}_i \in \mathbb{R}^{1\times T}$ (the i-th row of $\boldsymbol{\Psi}$, illustrated as a green histogram in the upper left corner). To simplify the subscripts of subsequent $\boldsymbol{\psi}$ variants, we temporarily drop the subscript of $\boldsymbol{\psi}_i$ as $\boldsymbol{\psi}$ in this section.

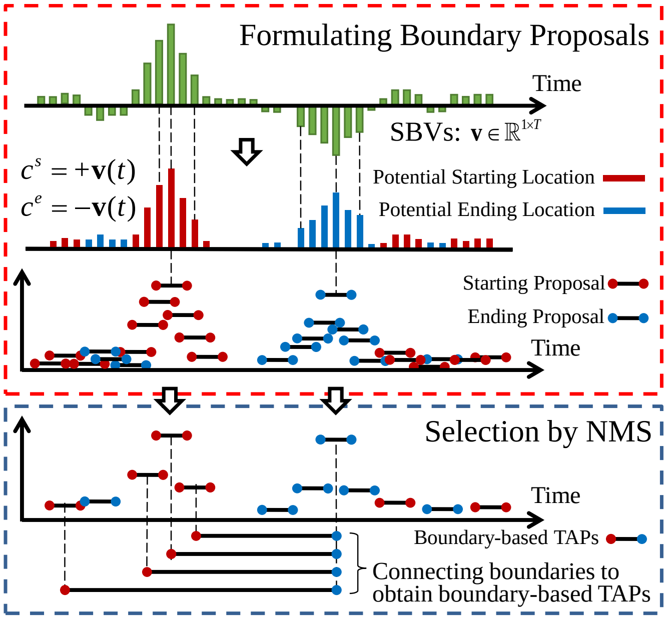

Formulating Boundary Proposals
Time
SBVs: $\mathbf{v} \in \mathbb{R}^{1\times T}$
$c^s = +\mathbf{v}(t)$
$c^e = -\mathbf{v}(t)$
Potential Starting Location
Potential Ending Location
Starting Proposal
Ending Proposal
Time
Selection by NMS
Time
Boundary-based TAPs
Connecting boundaries to obtain boundary-based TAPs

Fig. 5: The workflow of the boundary-based TAP generator. Taking SBVs ($\mathbf{v} \in \mathbb{R}^{1\times T}$) as the input, the first step is formulating boundary proposals according to Eq. (20) (red dashed box). Subsequently, locating staring and ending boundaries can be reformulated as selecting boundary proposals. As shown in the blue dashed box, boundary proposals are selected via NMS and boundary-based TAPs are finally obtained by connecting the center locations of starting and ending proposals. All boundary-based TAPs of a specific ending proposal are illustrated on the bottom.

To exploit local information, we propose the local boundary vector $\mathbf{b}(t) \in \mathbb{R}^{1\times T}$ as a central difference to capture the changes in SCPs:

$$\mathbf{b}(t) = \boldsymbol{\psi}(t+1) - \boldsymbol{\psi}(t-1), \quad t = 1, \dots, T. \tag{18}$$

To further leverage the temporal contextual information, the temporal contrast vector $\mathbf{c} \in \mathbb{R}^{1\times T}$ proposed in Eq. (12) is leveraged. Afterwards, we proceed to combine the local and contextual information by snippet-wise addition of $\mathbf{b}$ and $\mathbf{c}$ as $\mathbf{v} = \mathbf{b} + \mathbf{c}$, where $\mathbf{v} \in \mathbb{R}^{1\times T}$ denotes the final SBVs of the target action category.

3.4.2 Boundary-based TAP Generator

Taking SBVs ($\mathbf{v} \in \mathbb{R}^{1\times T}$) as the input, the proposed WS-TAP generation method can provide a set of $\mathbf{P}^b$ (*i.e.*, $\mathbb{P}^b$) with flexible durations and precise boundaries with a "select-and-connect" strategy as shown in Figure 5.

The key is the strategy of locating starting and ending boundaries of action instances without temporal annotations for training. Based on SBVs, we transform the localization of boundaries to a proposal selection task. Specifically, we formulate a starting boundary as a starting proposal as

$$\mathbf{P}^s = [x^s_{\text{st}}, x^s_{\text{ed}}, c^s], \tag{19}$$

where x^s_{st}, x^e_{st} and c^s are the indexes of the starting snippet, the index of the ending snippet and the confidence score of $\mathbf{P}^s$, respectively. Similarly, an ending proposal can be denoted as $\mathbf{P}^e = [x^e_{\text{st}}, x^e_{\text{ed}}, c^e]$. All the starting and ending proposals are collectively termed boundary proposals.

Unlike action instances which are often dramatically different in temporal durations, we empirically observe that the temporal durations of the action boundaries are concentrated, *i.e.*, $\|x^s_{\text{ed}} - x^s_{\text{st}}\| \approx \omega$, where ω is a constant representing the average duration (in "number-of-snippets") of such boundary proposals. We fix $\omega = 3$ since only the action scores within a three-snippet window are considered when calculating local information in Eq. (19).

The confidence scores of the boundary proposals are assigned according to SBVs ($\mathbf{v} \in \mathbb{R}^{1\times T}$). Specifically, for the t-th snippet, if its SBV is positive (*i.e.*, $\mathbf{v}(t) > 0$), it will be regarded as a potential starting location and the confidence score of the corresponding starting proposal (*i.e.*, c^s) will be assigned as $\mathbf{v}(t)$. If $\mathbf{v}(t) < 0$, the t-th snippet will be regarded as a potential ending location and the confidence score of the corresponding ending proposal (*i.e.*, c^e) will be assigned as $-\mathbf{v}(t)$. As a result, for the t-th snippet, its staring proposal $\mathbf{P}^s$ or ending proposal $\mathbf{P}^e$ is defined as,

$$\begin{aligned} x^s_{\text{st}} = x^e_{\text{st}} = t - 1,\ \ c^s = \mathbf{v}(t), \\ x^s_{\text{ed}} = x^e_{\text{ed}} = t + 1,\ \ c^e = -\mathbf{v}(t). \end{aligned} \tag{20}$$

The generation of boundary proposals is illustrated in Figure 5 within the dashed red box.

After the boundaries are formulated as proposals with confidence scores, the localization of boundaries is transformed to a proposal selection task, which is performed with the standard Non-Maximum Suppression (NMS) algorithm. After the boundary proposal selection, $\mathbb{P}^b$ is generated by connecting the selected starting and ending proposals, as shown in Figure 5 within the dashed blue box. Finally, the confidence scores of TAPs in $\mathbb{P}^b$ are also obtained leveraging the contrast score proposed in Section 3.3.2.

Note that our connecting step is the same as that in BSN [30]. However, BSN is fully supervised, and it can directly obtain starting and ending locations through training. Due to lack of temporal annotations, we cannot directly locate the starting and ending locations. This is why we transform the localization of boundaries to a proposal selection task. We argue that the greatest challenge is how to obtain the starting and ending locations with weak supervision, instead of how to connect them.

3.4.3 TAP Fusion

As claimed in [13], the TAPs generated by sliding windows and actionness-score-grouping methods are complementary. Similarly, $\mathbb{P}^b$ may miss real action instances if the boundaries are incorrectly located. Although $\mathbb{P}^w$ is not flexible enough, it can cover the whole input video. This inspires us to further fuse $\mathbb{P}^w$ with $\mathbb{P}^b$ to get the final TAP set $\mathbb{P}$. The simplest fusion practice is the direct set union, *i.e.*, $\mathbb{P} = \mathbb{P}^w \cup \mathbb{P}^b$, which is also discussed in our ablation study.

The goal of fusion is to exploit $\mathbb{P}^w$ to supplement $\mathbb{P}^b$ for better coverage. We use NMS with two overlap thresholds to achieve this goal. Specifically, denoting α as the NMS overlap threshold, $\forall \mathbf{P}^w \in \mathbb{P}^w$, it will be suppressed under either condition: (1) $\exists \mathbf{P} \in \mathbb{P}^w, s(\mathbf{P}) > s(\mathbf{P}^w)$ & $\text{IoU}(\mathbf{P}^w, \mathbf{P}) > \alpha$; (2) $\exists \mathbf{P} \in \mathbb{P}^b, s(\mathbf{P}) > s(\mathbf{P}^w)$ & $\text{IoU}(\mathbf{P}^w, \mathbf{P}) > \frac{1}{2}\alpha$. In this way, the final set $\mathbb{P}$ keeps all boundary-based proposals and meanwhile contains sliding-window-based proposals to supplement $\mathbb{P}^b$.

4 Training CleanNet

Having introduced the architecture of CleanNet, this section will discuss its training process. The action classification module is trained by minimizing the cross-entropy loss between the video-level prediction and the video-level categorical label, while the TAP regressor within the action localization module are optimized by maximizing the contrast scores of TAPs. Specifically, we first prove that the proposed TAP evaluator is differentiable, which guarantees the contrast score can be leveraged to train the regression model through the regular back-propagation algorithm. Afterwards, we describe the training scheme of the action localization and classification modules, and further present a joint finetuning process.

4.1 Back-propagation of the TAP Evaluator

Subsequently, we will prove that the proposed TAP evaluator is differentiable w.r.t. the input boundaries, *i.e.*, the contrast score $s(\mathbf{P})$ of an TAP $\mathbf{P}$ in Eq. (17) is differentiable w.r.t. its starting and ending boundaries x_s and x_e. To achieve this goal, we first define a score function as

$$s_o(x_1, x_2, \boldsymbol{\lambda}) = \frac{\int_{x_1}^{x_2} \boldsymbol{\lambda}(t)dt}{x_2 - x_1}, \tag{21}$$

where $[x_1, x_2]$ are the boundaries of a given TAP, and $\boldsymbol{\lambda} \in \mathbb{R}^{1\times T}$ is a snippet-level score. Intuitively, $s_o(x_1, x_2, \boldsymbol{\lambda})$ is the average score of given $\boldsymbol{\lambda}$ from x_1 to x_2. Afterwards, its gradient w.r.t. the starting boundary x_1 is derived as

$$\begin{aligned} \frac{\partial s_o(x_1, x_2, \boldsymbol{\lambda})}{\partial x_1} &= \frac{-\boldsymbol{\lambda}(x_1)(x_2 - x_1) + \int_{x_1}^{x_2} \boldsymbol{\lambda}(t)dt}{(x_2 - x_1)^2} \\ &= \frac{-\boldsymbol{\lambda}(x_1) + s_o(x_1, x_2, \boldsymbol{\lambda})}{(x_2 - x_1)}. \end{aligned} \tag{22}$$

Similarly, we can derivate the gradient w.r.t. the ending boundary x_2 as

$$\frac{\partial s_o(x_1, x_2, \boldsymbol{\lambda})}{\partial x_2} = \frac{\boldsymbol{\lambda}(x_2) - s_o(x_1, x_2, \boldsymbol{\lambda})}{(x_2 - x_1)}. \tag{23}$$

If the snippet-level score $\boldsymbol{\lambda} \in \mathbb{R}^{1\times T}$ is $\boldsymbol{\psi_i}$, $x_1 = x_s^{\text{def}}$ and $x_2 = x_e^{\text{def}}$, s_o will be s_a in Eq. (14). Therefore, s_a is differentiable w.r.t. x_s^{def} and x_e^{def} and the gradients can be calculated following Eq. (22-23). Similarly, s_s and s_e are differentiable w.r.t. their input boundaries and the gradients can also be calculated following Eq. (22-23). Finally, the whole contrast score $s(\mathbf{P})$ is differentiable w.r.t. x_s and x_e. Therefore, the pseudo-supervision provided by the contrast score can be leveraged to train the TAP regressor through the regular back-propagation algorithm.

4.2 Training of Action Localization and Classification

As shown in Figure 1, there are two losses, *i.e.*, the regression loss and the classification loss, which are responsible for the two outputs of CleanNet, *i.e.*, action localization and action classification, respectively.

Training of Action Localization. To train the action localization module, we first select *"positive"* TAPs according to their contrast scores assigned by the TAP evaluator. We adopt a similar proposal selection scheme as described in AutoLoc [43]. Specifically, when locating an action instance of the i-th category, if the i-th category prediction of the t-th snippet $\boldsymbol{\psi}_i(t)$ or its attention prediction $\boldsymbol{\varphi}(t)$ is lower than their corresponding pre-defined thresholds, all anchors centered at this snippet will be discarded. Afterwards, the remaining anchors are regressed to be sliding-window-based TAPs. An TAP $\mathbf{P}$ will be selected to be "positive", if its contrast score $s(\mathbf{P})$ is higher than 0.5. The set of all selected "positive" TAPs is denoted as $\mathbb{P}^p$. With $\mathbb{P}^p$, the regression loss L_{reg} is defined as

$$L_{\text{reg}} = -\frac{1}{|\mathbb{P}^p|}\sum_{\mathbf{P}\in\mathbb{P}^p} s(\mathbf{P}), \tag{24}$$

where $|\cdot|$ denotes the cardinality of a set (number of elements).

Training of Action Classification. The training process of the action classification module is the same as that of UntrimmedNet [53]. The classification loss is defined as

$$L_{\text{cls}} = \sum_{i=1}^{N} -\mathbf{y}(i)\log(\mathbf{p}(i)), \tag{25}$$

where $\mathbf{p}(i)$ and $\mathbf{y}(i)$ are the video-level prediction and the label of the i-th category. $\mathbf{p} \in \mathbb{R}^{N\times 1}$ is calculated as

$$\mathbf{p} = \sum_{t=1}^{T} \bar{\boldsymbol{\varphi}}(t)\boldsymbol{\Psi}(t), \tag{26}$$

where $\bar{\boldsymbol{\varphi}}$ is the attention weights of T snippets normalized by a softmax layer as

$$\bar{\boldsymbol{\varphi}}(t) = \frac{\exp(\boldsymbol{\varphi}(t))}{\sum_{t=1}^{T}\exp(\boldsymbol{\varphi}(t))}. \tag{27}$$

Therefore, L_{cls} is the cross-entropy loss between the video-level categorical label $\mathbf{y}$ and video-level categorical prediction $\mathbf{p}$. Intuitively, $\mathbf{p}$ is the weighted summation of all snippet-level predictions in the video, regardless of whether a snippet is background or not. In the case of videos with multiple labels, $\mathbf{y}$ will be normalized to have a unit ℓ_1-norm before training.

If the action classification module is initialized using pre-trained UntrimmedNet [53] as described in Section 3.2, we will skip the minimization of L_{cls}. And the finetuning of the action classification module is achieved by the joint finetuning process below.

Joint Finetuning Process. There is a drawback of training the action classification module by itself. All snippets are engaged in training regardless of whether they are background or not, which will inevitably introduce noise to the training procedure of action classification. Here we propose a simple yet effective way to further finetune the action classification module together with the action localization module (C_5 in Table 1). First, we find all snippet indexes covered by positive TAPs and denote this index set as $\mathbb{S}$. Intuitively, $\mathbb{S}$ is the set of all positive snippets. Afterwards, we propose a joint loss (L_{J}) to focus on the located action (by minimizing L_a) and suppress background (by minimizing L_b). They are defined as

$$L_{\text{J}} = L_a + L_b, \tag{28}$$

$$L_a = \sum_{i=1}^{N} -\mathbf{y}(i)\log(\mathbf{p}_a(i)), \tag{29}$$

$$L_b = \sum_{i=1}^{N} -\mathbf{y}(i)\log(1-\mathbf{p}_b(i)), \tag{30}$$

where the video-level predictions of action and background (*i.e.*, $\mathbf{p}_a$ and $\mathbf{p}_b$) are defined as

$$\mathbf{p}_a = \sum_{t\in\mathbb{S}}\hat{\boldsymbol{\varphi}}(t)\boldsymbol{\Psi}(t),\ \mathbf{p}_b = \frac{1}{T-|\mathbb{S}|}\sum_{t\notin\mathbb{S}}\boldsymbol{\Psi}(t). \tag{31}$$

$\hat{\boldsymbol{\varphi}}$ is the attention weights normalized by a softmax layer across snippets in $\mathbb{S}$, obtained by

$$\hat{\boldsymbol{\varphi}}(t) = \frac{\exp(\boldsymbol{\varphi}(t))}{\sum_{t\in\mathbb{S}}\exp(\boldsymbol{\varphi}(t))}. \tag{32}$$

Different from $\mathbf{p}$ in Eq. (26), $\mathbf{p}_a$ avoids the distraction from the irrelevant snippets because $\mathbb{S}$ contains only positive snippets. Therefore, minimizing L_a can make the classification module focus on the located action. On the contrary, $\mathbf{p}_b$ involves only snippets that are not contained by $\mathbb{S}$, which should be regarded as background snippets during the training of action classification. By minimizing L_b, background suppression can be achieved via recognizing background snippets as not the target action class.

The similar idea of background suppression is explored in [26], [35]. Different from these methods, our method uses localization results (*i.e.*, $\mathbb{S}$) to differentiate video-level action/background predictions (noted as $\mathbf{p}_a$ and $\mathbf{p}_b$), while [26] and [35] rely on snippet-level attentions. Therefore, this finetuning process actually leverages the localization results to finetune the classification module, making the action classification and localization benefit each other. Besides, compared with [26] and [35], no additional background class is introduced in our method.

5 Experiments

In this section, we evaluate the TAL performance of the proposed CleanNet, and carry out detailed ablation studies to explore the performance contribution of each component in CleanNet. Meanwhile, we compare our method with existing WS-TAL methods and recent fully-supervised TAL methods on three standard benchmarks.

5.1 Experimental Setting

Evaluation Datasets. The THUMOS14 [23] dataset contains 413 untrimmed videos of 20 actions in the temporal action localization (TAL) task, where 200 untrimmed videos form the validation set and 213 untrimmed videos form the testing set. Each video contains at least one action. The validation and testing sets are leveraged to train and evaluate our CleanNet, respectively. The training set of THUMOS14 is not related to the TAL task.

ActivityNet v1.2 & v1.3 [10] covers 100 & 200 activity categories. The training set includes $4,819$ & $9,997$ videos

TABLE 1: Five main components of CleanNet divided for detailed ablation studies.

Notation	Explanation
C_1	Training the TAP regressor.
C_2	Using s_a to evaluate TAPs.
C_3	Using s_s and s_e to evaluate TAPs.
C_4	Using TSN sampling strategy.
C_5	Joint finetuning of action classification.

TABLE 2: TAL performance comparison of our method and its variants with different combination of components on the THUMOS14 testing set, at the IoU threshold 0.5.

Method	C_1	C_2	C_3	C_4	C_5	mAP(%)
UntrimmedNet [53]	Baseline					15.4
Plain-Model		✓	✓			21.6
Actioness-Only	✓	✓		✓		1.2
Edgeness-Only	✓		✓	✓		11.4
CleanNet-Simple	✓	✓	✓			22.9
CleanNet-T	✓	✓	✓	✓		23.4
CleanNet-J	✓	✓	✓		✓	24.0
CleanNet-SW	✓	✓	✓	✓	✓	**24.3**

and the validation set includes $2,383$ & $4,998$ videos , which are used in our training and evaluation, respectively.

Evaluation Metrics. For the TAP generation task, Average Recall (AR) over multiple IoU thresholds at Average Number of proposals (AN), *i.e.*, AR@AN, is adopted as the evaluation metric. Following the conventional setting, we use IoU thresholds set [0.5:0.05:1.0] on THUMOS14 and [0.5:0.05:0.95] on ActivityNet. For example, AR@100 means average recall with 100 proposals. On ActivityNet, the area under the AR-vs.-AN curve (AUC) is also used as a metric, with the AN ranging from 0 to 100. For the TAL task, mean Average Precision (mAP) at multiple IoU thresholds, *i.e.*, mAP@IoU, is adopted as the evaluation metric, where Average Precision (AP) is calculated category-by-category. IoU thresholds are selected as [0.3:0.1:0.7] and [0.5:0.05:0.95] on THUMOS14 and ActivityNet, respectively.

Implementation details. We implement our CleanNet using PyTorch [37] on one NVIDIA GeForce GTX TITAN Xp GPU. We adopt stochastic gradient descent (SGD) solver for optimization. The training step composes two stages, the first/second stage minimizes $L_{\text{reg}}/L_{\text{J}}$, respectively. Both stages are with the initial learning rate of 0.0001 and divided by 10 after every 200 batches (one batch contains one whole untrimmed video). Following [43], the anchor sizes are set as 1, 2, 4, 8, 16, 32 snippets for THUMOS14 and 16, 32, 64, 128, 256, 512 snippets for ActivityNet, respectively. During testing for TAL task, NMS with IoU threshold $\alpha = 0.4$ is used to remove duplicated TAPs. For videos with multiple labels, we perform action localization to all actions with a classification score higher than 0.1. It takes 1/12/18 hours to train the models on THUMOS14/ActivityNet v1.2/v1.3. For the testing process, we leverage multithreaded programming to speed it up. Specifically, it takes around 6/42/62 minutes to test the models on THUMOS14/ActivityNet v1.2/v1.3.

5.2 Ablation Study

TABLE 3: TAL mAP on the THUMOS14 testing set, at the IoU threshold 0.5 with different pooling kernel sizes.

Method	Pooling Kernel Size				
	5	7	9	11	13
Plain-Model	21.1	21.6	21.9	21.8	21.3
CleanNet-SW	23.2	24.3	24.2	23.9	23.0

We present multiple ablation studies to explore the performance contribution of the two newly proposed modules, *i.e.*, the action localization module and the WS-TAP generation module.

5.2.1 Ablation Study on Action Localization Module

To explore the performance contribution of each component in action localization module, we first divide it into five components as listed in Table 1. Ablated variants with different combination of these five components are evaluated on THUMOS14, together with the baseline method UntrimmedNet [53], as presented in Table 2. Noting that to isolate the contribution of action localization module, the results in this subsection are obtained without the WS-TAP generation module.

Using Proposal Evaluator without Training TAP Regressor. Note that our TAP evaluator can assign contrast scores to an arbitrary TAP no matter whether it is regressed by the regressor or not. Therefore, with only the TAP evaluator in action localization, our CleanNet without training the TAP regressor can still function well. In this way, all "TAPs sampled via sliding windows without regression" (*i.e.*, anchors) are directly scored by the TAP evaluator. The rest steps remain the same. This ablated variant is denoted as "Plain-Model" in Table 2, since there is no trainable parameter for action localization. With such settings, action localization degenerates as a post-processing procedure and achieves a fair comparison with the thresholding component in UntrimmedNet [53]. The proposed TAP evaluator offers substantial improvement over UntrimmedNet [53] as the mAP is boosted from 15.4% to 21.6% at the IoU threshold 0.5. This ablation study validates the efficacy of the contrast scores provided by the TAP evaluator, which is responsible for the major improvement of our TAL performance.

Variants of Proposal Scores. As alternatives to the contrast score $s(\mathbf{P})$ defined in Eq. (17), two ablated versions are studied, termed "Actioness-Only" and "Edgeness-Only" in Table 2. The Actioness-Only replaces Eq. (17) with action score (C_2) only, *i.e.*, $s(\mathbf{P}) = s_a(\mathbf{P})$; while the Edgeness-Only replaces Eq. (17) with starting and ending scores (C_3) only, *i.e.*, $s(\mathbf{P}) = (s_s(\mathbf{P}) + s_e(\mathbf{P}))/2$.

As shown in Table 2, without $s_s(\mathbf{P})$ and $s_e(\mathbf{P})$, Actioness-Only suffers from such dramatic performance degradation that it performs significantly worse than UntrimmedNet [53]. The main reason of such severe degradation is that, $s_a(\mathbf{P})$ accounts for only content and ignores completeness, which results in assigning high scores to TAPs with poor boundaries. Considering only the boundaries, the performance of Edgeness-Only is marginally better than Actioness-Only, but the degradation is still obvious. This is because without considering the content (the action score), Edgeness-Only is likely to be more susceptible to fluctuations of SCPs (*e.g.*, due to noises). Comparing these two variants with others (both C_2 and C_3 are enabled), we

TABLE 4: TAL performance comparison between UntrimmedNet [53] and CleanNet-Simple on THUMOS14 test set, with the same feature embedding and SCPs.

Method	mAP@IoU						
	0.1	0.2	0.3	0.4	0.5	0.6	0.7
UntrimmedNet [53]	43.6	37.5	29.8	22.8	15.4	8.3	4.2
CleanNet-Simple	47.8	41.9	36.3	29.6	22.9	13.8	5.4

TABLE 5: TAL performance comparison between training with and without the pre-trained classification modules.

Initialization	Methods	mAP@IoU				
		0.3	0.4	0.5	0.6	0.7
W/ Pre-trained	CleanNet-Simple	36.3	29.6	22.9	13.8	5.3
W/o Pre-trained	CleanNet-Simple	36.7	29.8	23.0	13.6	5.2
W/ Pre-trained	CleanNet-SW	38.1	31.3	24.3	14.2	6.5
W/o Pre-trained	CleanNet-SW	38.3	31.3	24.4	14.1	6.5

provide performance advantage due to each term in Eq. (17), confirming $s_a(\mathbf{P})$, $s_s(\mathbf{P})$, and $s_e(\mathbf{P})$ are all indispensable components of the contrast score $s(\mathbf{P})$.

TSN Sampling and Joint Training. With components C_1, C_2 and C_3, the ablated version "CleanNet-Simple" in Table 2 has already achieved state-of-the-art performance, as presented in Table 9. This solidly validates our TAP evaluator contributes the most to our superior performance. Besides, enabling the TSN sampling ("CleanNet-T") or the joint finetuning of action classification ("CleanNet-J") can lead to further improvements over CleanNet-Simple. The comparison of CleanNet-T, CleanNet-J and CleanNet-SW shows that the contributions of C_4 and C_5 are compatible. Finally, with all five components enabled, CleanNet-SW achieves the best action localization performance among all variants.

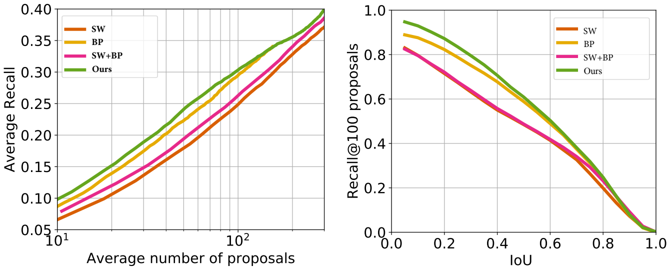
Sensitivity of Hyper-parameter Kernel Size. As shown in Table 2, the untrained Plain-Model is responsible for 70% performance gains over the baseline method UntrimmedNet [53]. Moreover, the Plain-Model provides a lower performance bound of CleanNet regardless of training configuration of the hyper-parameters, such as "positive threshold", learning rate, number of epochs, *etc*. However, the only parameter that affects the Plain-Model performance is the "pooling kernel size" when computing Eq. (10), and thus we present additional sensitivity tests in Table 3. Our method can work well under different pooling kernel sizes in general, which demonstrates the robustness of our method.

Overhead Analyses. The "nearest neighbor" of our action localization module is AutoLoc [43]. Compared with AutoLoc, the extra computation comes from the calculation of the contrast score and the classification model (FLOPs= T × 418K, where T is the number of snippets in the input video.). The former is responsible for the majority of the computational overhead since each proposal needs to be scored individually, while the latter only operates once per video. Specifically, AutoLoc computes mean values three times per proposal while CleanNet computes five times. For reference, taking UntrimmedNet [53] as the baseline, the overhead brought by the additional regression layers of AutoLoc is T × 890K FLOPs.

Comparison with Thresholding-based Methods. To further validate the advantage of our action localization over thresholding-based counterparts, we compare our method with thresholding-based methods. Specially, we highlight the comparison between UntrimmedNet [53] and CleanNet-Simple under the UNet backbone. Because they both share the same feature embedding and action classification modules, such comparison could reveal the effectiveness of our proposed action localization. AutoLoc [43] is also visualized for reference.

TABLE 6: AR@AN= $50, 100, 200$ performance comparison on THUMOS14 test set. Same ablated versions with different backbone networks are listed for comparison.

	Method	$\mathbb{P}^w$	$\mathbb{P}^b$	Fusion	@50	@100	@200
UNet	SW	✓			17.87	24.84	32.82
	BP		✓		22.29	29.49	-
	SW+BP	✓	✓		19.39	26.26	34.02
	Ours	✓	✓	✓	**24.09**	**30.28**	**36.71**
I3D	SW	✓			18.26	25.82	34.45
	BP		✓		**23.37**	30.32	-
	SW+BP	✓	✓		18.89	26.22	34.62
	Ours	✓	✓	✓	23.17	**30.88**	**37.09**



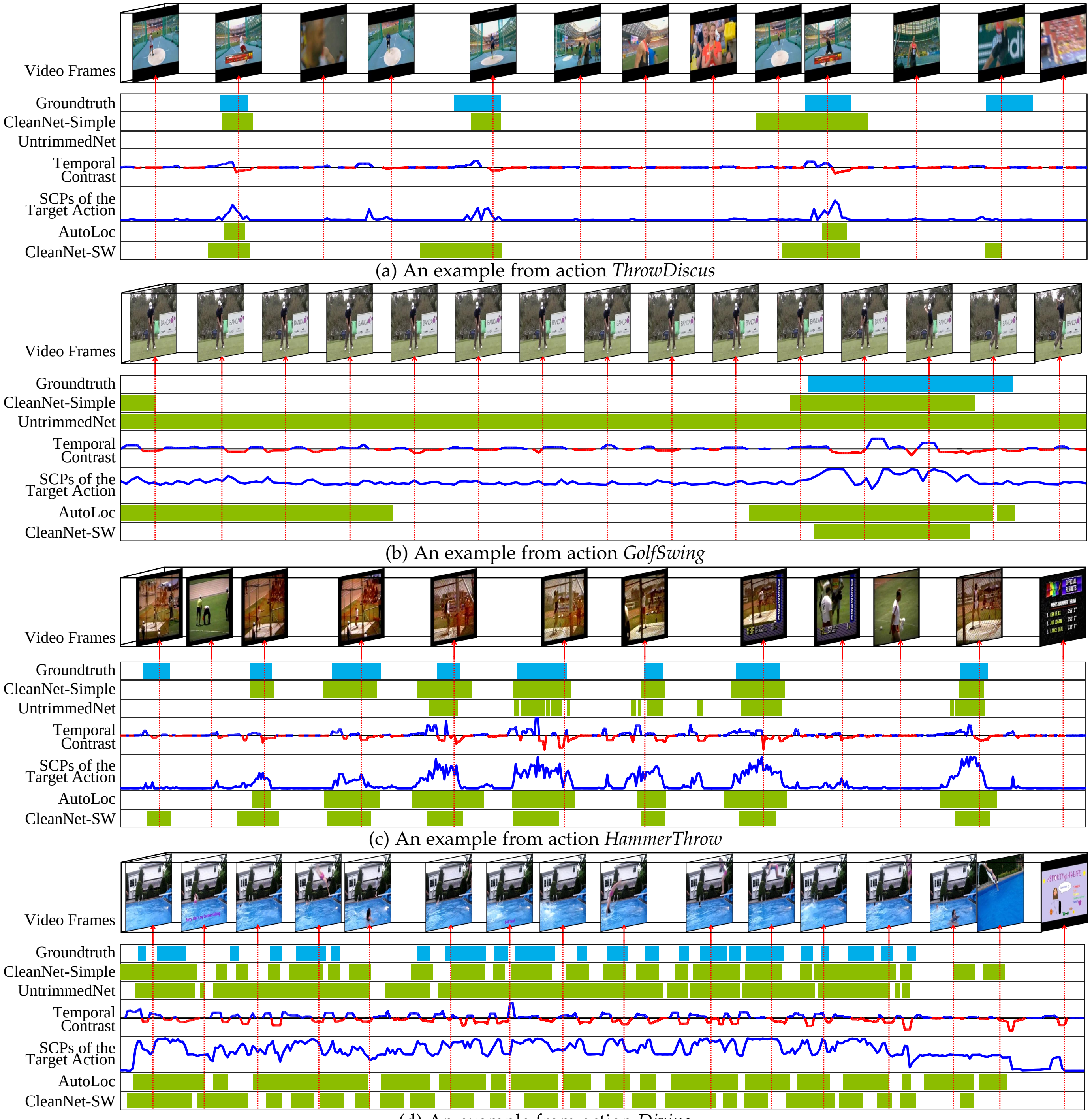
(a) (b)

Fig. 6: AR-vs.-AN and recall@AN=100-vs.-IoU comparison on THUMOS14 with UNet backbone.

Quantitatively, our method significantly outperforms UntrimmedNet [53] as shown in Table 4. Besides, we also present additional qualitative examples on THUMOS14 in Figure 7 to demonstrate our method can promote the completeness of TAPs. Note that the localization results of UntrimmedNet [53] and CleanNet-Simple are achieved with shared SCPs of the action.

Some challenging cases are illustrated in Figure 7 with a false negative error (Figure 7(a), *i.e.*, missing action instances) and a false positive error (Figure 7(b), *i.e.*, producing spurious action instances). Such errors are more prominent for UntrimmedNet [53], which could be caused by the difficulty of adjusting proper localization thresholds in UntrimmedNet. Other problems such as over-segmentation (*i.e.*, breaking one action instance into multiple ones) and under-segmentation (*i.e.*, merging multiple instances into one segment) are also generally more severe for UntrimmedNet [53], as illustrated in Figure 7(c) and Figure 7(d), respectively. We speculate that such a thresholding-based method only accounts for the content of TAPs but ignores specific treatment of proposal boundaries and context information.

Compared with thresholding-based methods, methods not relying on thresholding for localization can better handle these cases. Facilitated by the proposed TAP evaluator, our method can select TAPs considering both the content and the completeness, which could be the justification for its better performances.



(a) An example from action *ThrowDiscus*

(b) An example from action *GolfSwing*

(c) An example from action *HammerThrow*

(d) An example from action *Diving*

Fig. 7: Qualitative TAL examples of CleanNet-Simple, UntrimmedNet [53], AutoLoc [43] and CleanNet-SW on the THUMOS14 testing set. The ground truth temporal locations and predicted ones are illustrated with blue and green bars, respectively. Both the corresponding temporal contrast and snippet-level classification predictions (SCPs) of the target action are included. Specifically, for the temporal contrast, a two-tone color scheme is used, with blue and red colors representing positive and negative values, respectively. CleanNet-Simple and UntrimmedNet share the same SCPs. (a) An example video with false negative errors. (b) An example video with false positive errors. (c) An example of over-segmentation (*i.e.*, breaking one instance into multiple segments). (d) An example of under-segmentation (*i.e.*, merging multiple instances into one segment). Compared with the thresholding-based method UntrimmedNet, methods not relying on thresholding for localization can better handle these challenging cases.

TABLE 7: TAL: mAP@IoU comparison on THUMOS14 test set. Same methods with different backbone networks are listed for comparison.

	Method	mAP@IoU					Avg.
		0.3	0.4	0.5	0.6	0.7	
UNet	CleanNet-SW	38.1	31.3	24.3	14.2	6.5	22.9
	CleanNet-BP	36.8	31.4	**24.5**	14.9	6.6	22.9
	CleanNet	**38.2**	**31.5**	**24.5**	**15.0**	**7.4**	**23.3**
I3D	CleanNet-SW	43.6	33.8	25.0	14.2	7.0	25.1
	CleanNet-BP	43.6	**37.5**	**27.6**	**18.0**	**7.6**	**26.7**
	CleanNet	**44.4**	36.3	27.1	17.3	7.3	26.5

TABLE 8: Differences among the variants of our method.

Variants	Task	TAPs		Post-processing		Fusion
		$\mathbb{P}^w$	$\mathbb{P}^b$	$\mathbb{P}^w$	$\mathbb{P}^b$	
SW	TAP Gen.	✓		-	-	-
CleanNet-SW	WS-TAL	✓		NMS	-	-
BP	TAP Gen.		✓	-	-	-
CleanNet-BP	WS-TAL		✓	-	Single	-
SW+BP	TAP Gen.	✓	✓	-	-	-
Ours	TAP Gen.	✓	✓	-	-	✓
CleanNet	WS-TAL	✓	✓	-	Single	✓

TABLE 9: TAL performance comparison on the THUMOS14 testing set. Fully-supervised methods have access to both video-level category labels and temporal annotations during training; while the weakly-supervised methods only have video-level category labels.

	Method	mAP(%)@IoU				
		0.3	0.4	0.5	0.6	0.7
Fully-supervised	Wang *et al.* [52]	14.6	12.1	8.5	4.7	1.5
	S-CNN [44]	36.3	28.7	19.0	10.3	5.3
	CDC [42]	40.1	29.4	23.3	13.1	7.9
	Gao *et al.* [14]	50.1	41.3	31.0	19.1	9.9
	SSN [63]	51.9	41.0	29.8	19.6	10.7
	Chao *et al.* [4]	53.2	48.5	42.8	33.8	20.8
	BSN [30]	53.5	45.0	36.9	28.4	20.0
Weakly-supervised	Hide-and-Seek [47]	19.5	12.7	6.8	-	-
	UntrimmedNet [53]	29.8	22.8	15.4	8.3	4.2
	STPN [34]	31.1	23.5	16.2	9.8	5.1
	W-TALC [38]	32.0	26.0	18.8	10.9	6.2
	AutoLoc [43]	35.8	29.0	21.2	13.4	5.8
	CleanNet-Simple	36.3	29.6	22.9	13.8	5.3
	CleanNet	**38.2**	**31.5**	**24.5**	**15.0**	**7.4**
	STPN (I3D) [34]	35.5	25.8	16.9	9.9	4.3
	W-TALC (I3D) [38]	40.1	31.1	22.8	14.5	**7.6**
	CMCS (I3D) [31]	41.2	32.1	23.1	15.0	7.0
	MAAN (I3D) [61]	41.1	30.6	20.3	12.0	6.9
	CleanNet (I3D)	**44.4**	**36.3**	**27.1**	**17.3**	7.3

Training without Pre-trained Classification Modules. For direct comparison with thresholding-based methods, we initialized classification modules using pre-trained UntrimmedNet [53], to achieve shared SCPs (as described in Section 3.2). Special initialization is required because the TAP evaluator needs reasonable SCPs provided by the action classification module to score TAPs. Otherwise, minimizing Eq. (24) (*i.e.*, maximizing average contrast scores of positive TAPs) will become meaningless and $\mathbb{S}$ cannot filter noisy snippets as expected. Therefore, the parameters of action localization and classification modules cannot be jointly trained from scratch.

Without using pre-trained parameters, the action classification module is trained by minimizing L_{cls} using SGD solver with initial learning rate of 0.0001. The learning rate is divided by 10 after every 20 epochs. After 60 epochs, parameters of the action classification module is learned and the training of action localization and joint finetuning process stay the same as introduced before. As shown in Table 5, no obvious final performance difference is observed between the training with and without the pre-trained classification modules.

5.2.2 Ablation Study on WS-TAP Generation Module

The contribution of the WS-TAP generation module is reflected on both the TAP generation and WS-TAL tasks. We evaluate the effect of boundary-based TAPs on both tasks.

TAP Generation Task. We present the ablation study to compare the WS-TAP generation module with the sliding-window-based TAPs ($\mathbb{P}^w$) provided by the TAP regressor with the TAP generation task. To isolate the contributions of the boundary-based TAPs ($\mathbb{P}^b$) generated in Section 3.4.2 and the TAP fusion scheme in Section 3.4.3, ablated variants of our WS-TAP generator are evaluated.

The AR@AN comparison on THUMOS14 test set is summarized in Table 6 and visualized in Figure 6(a). With only $\mathbb{P}^w$ as the baseline performance, both $\mathbb{P}^b$ and the fusion scheme contributes to the TAP generation performance. Besides, Recall@AN=100-vs.-IoU performance is illustrated in Figure 6(b), which implies the quality (*i.e.*, temporal overlap with ground truth) of the generated proposals. Our proposed method achieves significant higher Recall@AN=100 than other variants through a wide IoU range $0.1 \sim 0.7$.

WS-TAL Task. The contributions of boundary-based TAP for the WS-TAL task are compared in Table 7. Different from the TAP generation task, only a single TAP with the highest confidence score is kept at each temporal position (denoted as "Single" in Table 8). The ablated versions with only boundary-based TAPs are denoted as CleanNet-BP.

CleanNet-BP achieves better mAP than CleanNet-SW when IoU threshold is larger, which verifies our assumptions that boundary-based TAPs do generate high quality proposals, but not enough of them to cover all action instances, especially with low IoU thresholds. By combining them with the fusion scheme proposed in Section 3.4.3, the TAL performance is further improved compared with CleanNet-SW, which is our complete version for TAL task in Table 9 and Table 12. Evaluated on different tasks, the variants of our method have similar names in Table 6 and 7. For clarity, the differences among these similar variants are summarized in Table 8.

5.3 Performance Comparison with State-of-the-Arts

Experiment on THUMOS14 on TAL task. As summarized in Table 9, our method with a UNet backbone outperforms all the compared WS-TAL methods on the THUMOS14 testing set. The performance advantage of CleanNet is especially significant compared with thresholding-based methods, *e.g.*, Hide-and-Seek [47], UntrimmedNet [53], STPN [34], and W-TALC [38], which implies the superiority of the TAP regression and evaluation scheme over

TABLE 10: AR@AN performance comparison on THUMOS14 test set. Methods with full supervision use both video-level category labels and temporal annotations during training; while our method only have access to video-level category labels.

Method	Supervision	@50	@100	@200
TAG [63]	Full	18.55	29.00	39.61
TURN [15]	Full	21.86	31.89	43.02
CTAP [13]	Full	32.49	42.61	51.97
BSN+NMS [30]	Full	35.41	43.55	52.23
Ours	Weak	24.09	30.28	36.71
Ours (I3D)	Weak	23.17	30.88	37.09

TABLE 11: AUC and AR@100 comparisons on the validation set of ActivityNet v1.3. Our method is the only weakly supervised one.

Method	TAG	TURN	CTAP	BSN	**Ours**
	[63]	[15]	[13]	[30]	
AUC (val)	53.02	53.92	65.72	66.17	44.32
AR@100	63.52	63.46	73.17	74.16	50.35

thresholding. Moreover, CleanNet-Simple can be regarded as a direct comparison to AutoLoc [43], since it differs from AutoLoc only in TAP evaluator. Thanks to all the distinct designs (see Section 2.4 for details) of CleanNet, it outperforms AutoLoc at all IoU thresholds. Surprisingly, our method even achieves comparable performances with some fully-supervised TAL methods.

As shown in the bottom part of Table 9, our method with the I3D backbone [3] pretrained on Kinetics (Ours-I3D) also achieves state-of-the-art performance, compared with other WS-TAL methods with the same I3D backbone. This clearly manifests that our method is not tied to a specific backbone.

Experiment on ActivityNet v1.2 & v1.3 on TAL task. As the comparison results on ActivityNet v1.2 & v1.3 in Table 12 shown , our methods outperforms all other WS-TAL methods on average mAP at IoU thresholds 0.5:0.05:0.95 using I3D backbone. Taking W-TALC [38] to represent thresholding-based methods, it can achieve fair performance when the IoU threshold is lower. Owing to the low noise ratio of ActivityNet v1.2 validation set, which has only an average of 1.5 action instances and 34.6% background per video. In contrast, THUMOS14 has an average of 15.4 action instances and 71.4% background per video. However, our method keeps the highest performance at all IoU thresholds, which verifies that our method can select and keep the TAPs having larger overlaps with ground truth temporal action instances.

Experiment on TAP Generation task. As shown in Table 10, on the test set of THUMOS14, our method achieves competitive temporal proposal generation results against fully supervised approaches. As the only weakly supervised one, our method achieves competitive AR at small AN, indicating that our top few (within the small AN values) proposals are comparable in quality with those generated by fully supervised approaches. But our AR saturates faster as AN increases and there is a large performance gap between our method and fully supervised ones at AN=200, especially with non-sliding-window-based methods [13], [30], [63].

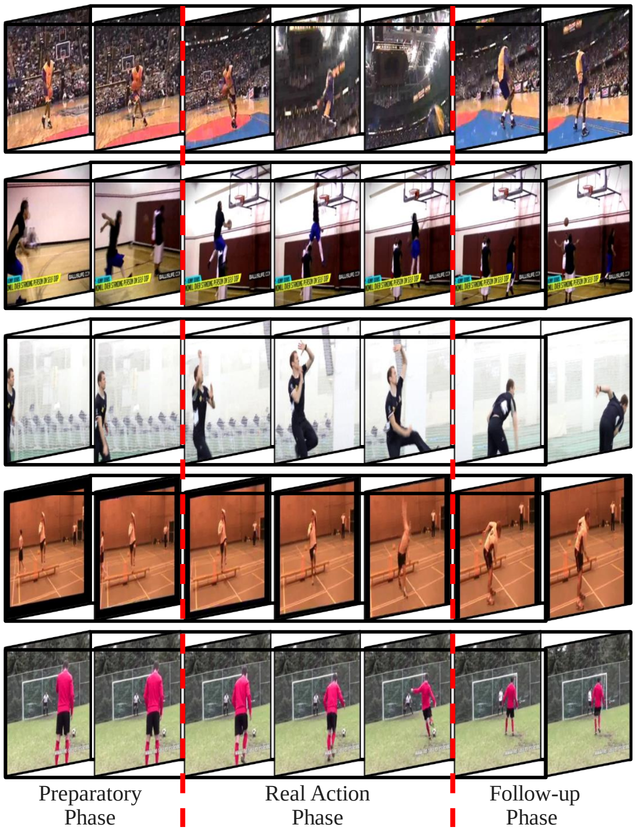


Fig. 8: Action instances with ambiguous boundaries, such as the run-up as the preparatory phase. With the immediately preceding preparatory phase and the subsequent follow-up phase, it is challenging for algorithms to precisely locate the real action phase, especially with weakly supervised methods. The above five samples demonstrate such cases, where our proposed method misclassifies these transitional phases as part of the real action instance. The dashed red lines indicate the real temporal action boundaries provided by the groundtruth.

As summarized in Table 11, on the validation set of ActivityNet v1.3, our method also achieves competitive TAP generation performance against fully supervised approaches. But the performance gaps between our method and fully supervised methods are larger than those on the THUMOS14 test set.

5.4 Strengths and Limitations.

For strengths, the experimental results show that our method (1) is capable of locating the action instances in untrimmed videos with only video-level labels during training, (2) is not tied to specific feature backbone, (3) has a more flexible localization module compared with thresholding-based methods, (4) can even achieve comparable performances with some fully-supervised methods. (5) the action localization module and action classification module can be jointly finetuned to achieve better performance. Besides, (6) we provide a WS-TAP generation method shows efficiency on both WS-TAL and TAP generation tasks.

For limitations, the first limitation of the CleanNet is that the parameters of action localization and classification

TABLE 12: TAL performance comparison on ActivityNet v1.2 and v1.3 validation set, in terms of mAP at IoU thresholds $[0.5 : 0.05 : 0.95]$. Our result is also comparable to fully-supervised models.

Supervision	Method	1.2/1.3	mAP(%)@IoU										Avg
			0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	
Full	SSN [63]	v1.2	41.3	38.8	35.9	32.9	30.4	27.0	22.2	18.2	13.2	6.1	26.6
Full	SSN [63]	v1.3	39.1	-	-	-	-	23.5	-	-	-	5.5	24.0
Weak	UntrimmedNet [53]	v1.2	7.4	6.1	5.2	4.5	3.9	3.2	2.5	1.8	1.2	0.7	3.6
Weak	AutoLoc [43]	v1.2	27.3	24.9	22.5	19.9	17.5	15.1	13.0	10.0	6.8	3.3	16.0
Weak	TSM [59]	v1.2	28.3	26.0	23.6	21.2	18.9	17.0	14.0	11.1	7.5	3.5	17.1
Weak	W-TALC [38]	v1.2	37.0	33.5	30.4	25.7	14.6	12.7	10.0	7.0	4.2	1.5	18.0
Weak	CMCS [31]	v1.2	36.8	-	-	-	-	22.0	-	-	-	**5.6**	22.4
Weak	**CleanNet**	v1.2	**40.5**	**35.9**	**32.5**	**28.8**	**25.5**	**22.3**	**18.7**	**14.8**	**9.8**	5.2	**23.4**
Weak	STPN [34]	v1.3	29.3	-	-	-	-	16.9	-	-	-	2.6	-
Weak	TSM [59]	v1.3	30.3	-	-	-	-	19.0	-	-	-	**4.5**	-
Weak	BM [35]	v1.3	36.4	-	-	-	-	19.2	-	-	-	2.9	-
Weak	**CleanNet**	v1.3	**36.7**	33.1	30.1	26.9	23.2	**20.4**	16.3	13.3	9.0	**4.5**	**21.4**

modules cannot be jointly trained from scratch, as discussed in Section 5.2.1. Another limitation is that although our method achieves state-of-the-art WS-TAL performance on average, it can still perform worse than UntrimmedNet [53] on a few categories such as *BasketballDunk*, *CricketBowling* and *SoccerPenalty*, where the action boundaries are ambiguous, as illustrated in Figure 8. CleanNet has difficulty in distinguishing the preparatory and follow-up phases from the real action phase with only video-level categorical labels available during training. We speculate that it might be necessary to incorporate temporal supervision (*i.e.*, full supervision) to handle these challenging action categories.

To summarize, on the WS-TAL task, our method achieves state-of-the-art WS-TAL performance on THUMOS14, ActivityNet v1.2 and v1.3 datasets. It can even achieve performances comparable to some fully-supervised methods. Moreover, extensive experiments in the ablation study provide some insights on the performance contribution of each component in CleanNet. On the TAP generation task, our method achieves competitive performance compared with fully supervised methods, facilitated by the extended WS-TAP generation module. The boundary-based TAPs generated by the WS-TAP generation module contributes most to the improvement and the TAP fusion module function well in supplementing to boundary-based TAPs especially in low IoU range.

6 Conclusion

We propose CleanNet for weakly-supervised temporal action localization, which leverages the temporal contrast among snippet-level action classification predictions to locate the temporal action boundaries. The new TAP evaluator provides contrast scores as pseudo-supervision to replace manually labeled temporal boundaries. Besides, we propose a new WS-TAP generation module compatible with weak supervision and introduce the concept of boundary proposals, which are located and evaluated by the SBVs we proposed. Boundary-based TAPs are then obtained by connecting boundary proposals to accommodate flexible durations. Combining the sliding-window-based TAPs and boundary-based TAPs, our method achieves state-of-the-art performance on both TAP generation and WS-TAL task.

Acknowledgments

This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 62088102, 61773312, and 61976171, Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001, and Natural Science Foundation of Shaanxi Grant 2020JQ-069.

References

[1] M. Asadiaghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Poncelopez, X. Baro, I. Guyon, S. Kasaei, and S. Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *Proc. IEEE Int. Conf. Auto. Face Gesture Recog.*, pages 476–483, 2017.

[2] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6373–6382, 2017.

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4724–4733, 2017.

[4] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1130–1139, 2018.

[5] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen. Temporal context network for activity localization in videos. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 5727–5736, 2017.

[6] O. Dan, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1817–1824, 2013.

[7] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars. Online action detection. In *Proc. Eur. Conf. Comput. Vis.*, pages 269–284, 2016.

[8] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, pages 65–72, 2005.

[9] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *Proc. Eur. Conf. Comput. Vis.*, pages 768–784, 2016.

[10] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 961–970, 2015.

[11] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 6202–6211, 2019.

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1933–1941, 2016.

[13] J. Gao, K. Chen, and R. Nevatia. Ctap: Complementary temporal action proposal generation. In *Proc. Eur. Conf. Comput. Vis.*, pages 68–83, 2018.

[14] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. In *Proc. Br. Mach. Vis. Conf.*, 2017.

[15] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3628–3636, 2017.

[16] R. Girshick. Fast r-cnn. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1440–1448, 2015.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 580–587, 2014.

[18] M. Hoai and F. De la Torre. Max-margin early event detectors. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2863–2870, 2012.

[19] L. Huang, Y. Huang, W. Ouyang, and L. Wang. Relational prototypical network for weakly supervised temporal action localization. In *Proc. AAAI Conf. Artif. Intell.*, pages 11053–11060, 2020.

[20] N. Hussein, E. Gavves, and A. W. Smeulders. Timeception for complex action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 254–263, 2019.

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Mach. Learn.*, pages 448–456, 2015.

[22] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, 2013.

[23] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[24] S. M. Kang and R. P. Wildes. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906*, 2016.

[25] I. Laptev. On space-time interest points. *Int. J. Comput. Vis.*, 64(2-3):107–123, 2005.

[26] P. Lee, Y. Uh, and H. Byun. Background suppression network for weakly-supervised temporal action localization. In *Proc. AAAI Conf. Artif. Intell.*, pages 11320–11327, 2020.

[27] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, and J. Song. Graph attention based proposal 3d convnets for action detection. In *Proc. AAAI Conf. Artif. Intell.*, pages 4626–4633, 2020.

[28] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, and R. Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proc. AAAI Conf. Artif. Intell.*, pages 11499–11506, 2020.

[29] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3889–3898, 2019.

[30] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proc. Eur. Conf. Comput. Vis.*, pages 3–19, 2018.

[31] D. Liu, T. Jiang, and Y. Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1298–1307, 2019.

[32] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei. Gaussian temporal awareness networks for action localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 344–353, 2019.

[33] K. Min and J. J. Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *Proc. Eur. Conf. Comput. Vis.*, pages 283–299, 2020.

[34] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 6752–6761, 2018.

[35] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes. Weakly-supervised action localization with background modeling. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 5502–5511, 2019.

[36] D. Oneata, J. Verbeek, and C. Schmid. The lear submission at thumos 2014. In *Proc. Eur. Conf. Comput. Vis. THUMOS Challenge*, 2014.

[37] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Proc. Adv. Neural Inf. Process. Syst. Worshop*, 2017.

[38] S. Paul, S. Roy, and A. K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proc. Eur. Conf. Comput. Vis.*, pages 588–607, 2018.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 779–788, 2016.

[40] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[41] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1234–1241, 2012.

[42] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1417–1426, 2017.

[43] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proc. Eur. Conf. Comput. Vis.*, pages 154–171, 2018.

[44] Z. Shou, D. Wang, and S. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1049–1058, 2016.

[45] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 568–576, 2014.

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learn. Rep.*, pages 521—534, 2015.

[47] K. K. Singh and Y. J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3524–3533, 2017.

[48] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proc. ACM Int. Conf. Multimedia*, pages 371–380, 2015.

[49] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4597–4605, 2015.

[50] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4489–4497, 2015.

[51] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3551–3558, 2013.

[52] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2):2, 2014.

[53] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4325–4334, 2017.

[54] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. Eur. Conf. Comput. Vis.*, pages 20–36, 2016.

[55] X. Wang and A. Gupta. Videos as space-time region graphs. In *Proc. Eur. Conf. Comput. Vis.*, pages 399–417, 2018.

[56] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 284–293, 2019.

[57] H. Xu, A. Das, and K. Saenko. R-c3d: region convolutional 3d network for temporal activity detection. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 5794–5803, 2017.

[58] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10156–10165, 2020.

[59] T. Yu, Z. Ren, Y. Li, E. Yan, N. Xu, and J. Yuan. Temporal structure mining for weakly supervised action detection. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 5522–5531, 2019.

[60] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3093–3102, 2016.

[61] Y. Yuan, Y. Lyu, X. Shen, I. W. Tsang, and D.-Y. Yeung. Marginalized average attentional network for weakly-supervised learning. In *Proc. Int. Conf. Learn. Rep.*, 2019.

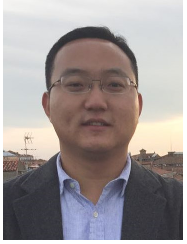
[62] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan. Graph convolutional networks for temporal action lo-

calization. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 7094–7103, 2019.

[63] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2933–2942, 2017.

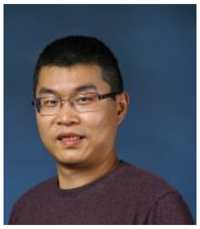
[64] L. Ziyi, W. Le, Z. Qilin, G. Zhanning, N. Zhenxing, Z. Nanning, and H. Gang. Weakly supervised temporal action localization through contrast based evaluation networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3899–3908, 2019.

Ziyi Liu received the B.S. degree in Control Science and Engineering from Xi'an Jiaotong University in 2015. He is currently a Ph.D. student with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. From 2019 to 2020, he was a visiting Ph.D. student with State University of New York at Buffalo. His research interests include computer vision and machine learning. He is a student member of the IEEE.



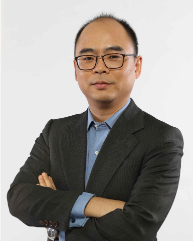
Le Wang (M'14-SM'20) received the B.S. and Ph.D. degrees in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with the Stevens Institute of Technology, Hoboken, New Jersey, USA. From 2016 to 2017, he was a visiting scholar with Northwestern University, Evanston, Illinois, USA. He is currently an Associate Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision, pattern recognition, and machine learning. He is the author of more than 50 peer reviewed publications in prestigious international journals and conferences. He is a senior member of the IEEE.

Qilin Zhang (M'18) received the B.E. degree in Electrical Information Engineering from the University of Science and Technology of China, Hefei, China, in 2009, the M.S. degree in Electrical and Computer Engineering from University of Florida, Gainesville, Florida, USA in 2011, and the Ph.D. degree in Computer Science from Stevens Institute of Technology, Hoboken, New Jersey, USA, in 2016. He is currently a Senior Research Scientist at ABB Corporate Research Center in Raleigh, North Carolina, USA. His research interests include computer vision, machine learning, multimedia signal processing, and autonomous driving. He is the author of over 20 peer reviewed publications in international journals and conferences. He is a member of the IEEE.



Wei Tang received his Ph.D. degree in Electrical Engineering from Northwestern University, Evanston, Illinois, USA in 2019. He received the B.E. and M.E. degrees from Beihang University, Beijing, China, in 2012 and 2015 respectively. He is currently an Assistant Professor in the Department of Computer Science at the University of Illinois at Chicago. His research interests include computer vision, pattern recognition and machine learning.

Nanning Zheng (SM'94-F'06) graduated in 1975 from the Department of Electrical Engineering, Xi'an Jiaotong University (XJTU), received the ME degree in Information and Control Engineering from Xi'an Jiaotong University in 1981, and a Ph. D. degree in Electrical Engineering from Keio University in 1985. He is currently a Professor and the director with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, and hardware implementation of intelligent systems. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy Engineering in 1999. He is a fellow of the IEEE.



Gang Hua (M'03-SM'11-F'19) was enrolled in the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU), Xi'an, China, in 1994 and received the B.S. degree in Automatic Control Engineering from XJTU in 1999. He received the M.S. degree in Control Science and Engineering in 2002 from XJTU, and the Ph.D. degree in Electrical Engineering and Computer Science at Northwestern University, Evanston, Illinois, USA, in 2006. He is currently the Vice President and Chief Scientist of Wormpex AI Research. Before that, he served in various roles at Microsoft (2015-18) as the Science/Technical Adviser to the CVP of the Computer Vision Group, Director of Computer Vision Science Team in Redmond and Taipei ATL, and Principal Researcher/Research Manager at Microsoft Research. He was an Associate Professor at Stevens Institute of Technology (2011-15). During 2014-15, he took an on leave and worked on the Amazon-Go project. He was an Visiting Researcher (2011-14) and a Research Staff Member (2010-11) at IBM Research T. J. Watson Center, a Senior Researcher (2009-10) at Nokia Research Center Hollywood, and a Scientist (2006-09) at Microsoft Live labs Research. He is an associate editor of TIP, TCSVT, CVIU, IEEE Multimedia, TVCJ and MVA. He also served as the Lead Guest Editor on two special issues in TPAMI and IJCV, respectively. He is a program chair of ICCV'2025. He is a program chair of CVPR'2019&2022. He is an area chair of CVPR'2015&2017, ICCV'2011&2017, ICIP'2012&2013&2016, ICASSP'2012&2013, and ACM MM 2011&2012&2015&2017. He is the author of more than 160 peer reviewed publications in prestigious international journals and conferences. He holds 19 US patents and has 15 more US patents pending. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award for his contribution on Unconstrained Face Recognition from Images and Videos, and a recipient of the 2013 Google Research Faculty Award. He is an IEEE Fellow, an IAPR Fellow, and an ACM Distinguished Scientist.