

# WOAD: Weakly Supervised Online Action Detection in Untrimmed Videos

Mingfei Gao, Yingbo Zhou, Ran Xu, Richard Socher, Caiming Xiong  
Salesforce Research

{mingfei.gao,yingbo.zhou,ran.xu,rsocher,cxiong}@salesforce.com

## Abstract

Online action detection in untrimmed videos aims to identify an action as it happens, which makes it very important for real-time applications. Previous methods rely on tedious annotations of temporal action boundaries for training, which hinders the scalability of online action detection systems. We propose WOAD, a weakly supervised framework that can be trained using only video-class labels. WOAD contains two jointly-trained modules, i.e., temporal proposal generator (TPG) and online action recognizer (OAR). Supervised by video-class labels, TPG works offline and targets at accurately mining pseudo frame-level labels for OAR. With the supervisory signals from TPG, OAR learns to conduct action detection in an online fashion. Experimental results on THUMOS'14, ActivityNet1.2 and ActivityNet1.3 show that our weakly-supervised method largely outperforms weakly-supervised baselines and achieves comparable performance to the previous strongly-supervised methods. Beyond that, WOAD is flexible to leverage strong supervision when it is available. When strongly supervised, our method obtains the state-of-the-art results in the tasks of both online per-frame action recognition and online detection of action start.

## 1. Introduction

Temporal Action Localization aims to detect temporal action boundaries in long, untrimmed videos. Most previous methods are under offline settings [2, 4, 5, 10, 25, 31], where they can observe the entire action before making decisions. However, applications such as surveillance systems and autonomous cars, are required to interact with the world in real time based on their accumulative observations up to now. Online Action Detection [6] is proposed to address this problem, where methods need to identify occurring actions moment-to-moment without access to future information. With different focuses, recent online action detectors consider two sub-tasks: (1) online per-frame action recognition [6, 9, 28] and (2) online detection of action start [11, 24]. The former task focuses on the general capability of recognizing the action category of each coming frame. On the other hand, detecting



Figure 1. Comparison between previous methods and our approach. Previous methods require **segment-level** annotations (start and end times of actions) during training, which leads to high human labeling cost. In contrast, our method can be trained with **video-level** annotations (video-level action labels) and is flexible to utilize (**full/partial**) segment-level supervision when it is available.

action starts in a timely manner is more important to some real-world applications. For example, an autonomous car needs to recognize “line merging” of another vehicle as soon as it starts. While, it is challenging to detect action starts due to the similar appearances near the start points and the lack of training data. The later task specially targets on this problem. Our method jointly addresses these two tasks.

Although previous methods have achieved promising progress, they rely on segment-level annotations of action boundaries for training (see Fig. 1). However, annotating action boundaries in long, untrimmed videos involves possibly ambiguous decisions and requires significant amount of human labor. This hinders the scalability of model learning, particularly for videos embodying complex semantics. Compared to the segment-level boundaries, video-level action classes are much easier to acquire. With the help of text-based video retrieval techniques, video-class labels may be obtained almost for free from the internet at a large scale.

To take advantage of the easy-to-obtain video-level annotations, we propose WOAD, a **Weakly supervised Online Action Detection** framework, that *can be trained with video-class labels only*. Detecting actions using weak supervision in an online scenario is challenging since (1) online action detectors generally require per-frame labels for training, so it is hard to utilize video-level labels as supervision and (2) it is not trivial for a model to be accurate for action recognition and sensitive to action starts without access to future information. As shown in Fig. 2, our proposed WOAD contains

two jointly-trained modules, i.e., Temporal Proposal Generator (TPG) and Online Action Recognizer (OAR), each of which focuses on handling one of the challenges. Supervised by video-class labels, TPG mines class-wise temporal action proposals that can be used as pseudo per-frame labels for OAR. While, OAR aims at conducting both per-frame action recognition and start detection jointly in an online fashion.

The proposed design has the following benefits: (1) TPG is used only during training for pseudo labels generation, so it can fully utilize temporal relation of frames (e.g. grouping nearby frames of the same class to improve proposal generation) without online constraint; (2) the design of OAR directly targets at improving the online tasks without being distracted by the weakly supervised setting and (3) the joint training could help learn better representations.

Our contributions are summarized as follows: (1) we introduce a novel method for weakly supervised online action detection. To the best of our knowledge, this is the first work that addresses the problem using weak supervision; (2) our method is flexible to combine weak and strong supervision when only a part of videos have strong annotations and (3) experimental results show that our method largely outperforms weakly-supervised baselines and achieves comparable performance to the previous strongly-supervised methods. When strongly supervised, our method obtains the state-of-the-art results in the tasks of both online per-frame action recognition and online detection of action start.

## 2. Related Work

**Temporal Action Localization.** The goal of temporal action detectors is to find the temporal boundaries of actions in untrimmed videos. Most existing methods work in offline settings, where they can make predictions after processing the entire actions. Shou et al. proposed S-CNN [25] to conduct action proposal generation, classification and regression via multi-stage networks. Dai et al. introduced TCN [5] that utilizes temporal context of proposals to improve proposal generation. Xu et al. presented R-C3D [27] that improves model efficiency by sharing the processing stages of proposal generation and classification. Buch et al. proposed SST [2] to conduct fast proposal generation. Zeng et al. modeled relations among proposals using Graph Convolutional Networks [14] and improved feature representations in [30].

**Online Action Detection.** Online action detectors identify the occurring action in untrimmed, streaming videos based on the past and current observations. Geest et al. first posed this problem as online per-frame action recognition and set up several baselines and evaluation metrics in [6]. Following this direction, Gao et al. introduced RED [9] which conducts current and future action predictions jointly. Xu et al. proposed TRN [28] that uses the predicted future actions to improve action recognition at the current time. Eun et al. introduced IDU to accumulate input information

based on its relevance to the current action in [7]. Compared to per-frame action recognition, online detecting action starts is more important for some applications and is more challenging due to the similar appearance near starts and the lack of training data. Shou et al. first proposed an online framework in [24] and treated the problem as a classification task. Gao et al. presented StartNet [11] which set the new state-of-the-art performance. However, these methods depend on the annotations of action boundaries for training and are evaluated on either per-frame action recognition or action start detection. Our work jointly handles these two tasks using weak supervision.

**Weakly Supervised Offline Action Detection.** Extensive studies have been done in offline action detection with video-class labels as supervision. Wang et al. introduced UntrimmedNet [26] to model actions from untrimmed videos. Shou et al. improved UntrimmedNet by introducing Outer-Inner-Contrastive loss [23]. Using only video-level labels, W-TALC [22] learns action representations using MIL and co-activity similarity losses. Liu et al. focused on the completeness of actions in [17] and BasNet [15] improved weakly supervised action localization by background suppression. In [20], Narayan et al. optimized the models by jointly minimizing category, count and center losses. Yuan et al. proposed MAAN [29] to relieve the effect of the dominant response of the most salient regions. Luo et al. explicitly modeled the key instances assignment via a EM-MIL approach in [18]. Nguyen et al. learned a rich notion of actions via background modeling in [21]. Min et al. proposed A2CL-PT [19] to learn discriminative features and distinguish background. Existing offline methods are not well suitable in the online setting since (1) offline methods target at predicting action segments that have significant temporal overlap with ground-truth, so they are not designed to be good at per-frame recognition and are not sensitive to start; (2) most of them adopt temporal prediction grouping strategy to improve performance during inference which violates online constraint and (3) technically, they prefer feedforward networks while online action detectors work better with RNNs.

## 3. Weakly Supervised Online Action Detection

### 3.1. Framework Overview

For online action detection, the input to the system is a streaming, untrimmed video,  $\mathbf{V}_i$ , represented as a sequence of image frames  $[I_i^1, I_i^2, \dots, I_i^{T_i}]$ , where  $i$  denotes video index and  $T_i$  is video length. At each time  $t$ , the system takes  $I_i^t$  as input. It predicts,  $\mathbf{a}_i^t$ , the probability of the current action category (online per-frame action recognition) and determines,  $\mathbf{as}_i^t$ , the probability that an action start occurs (online detection of action start). Under the constraint of online setting, no future information is available in the

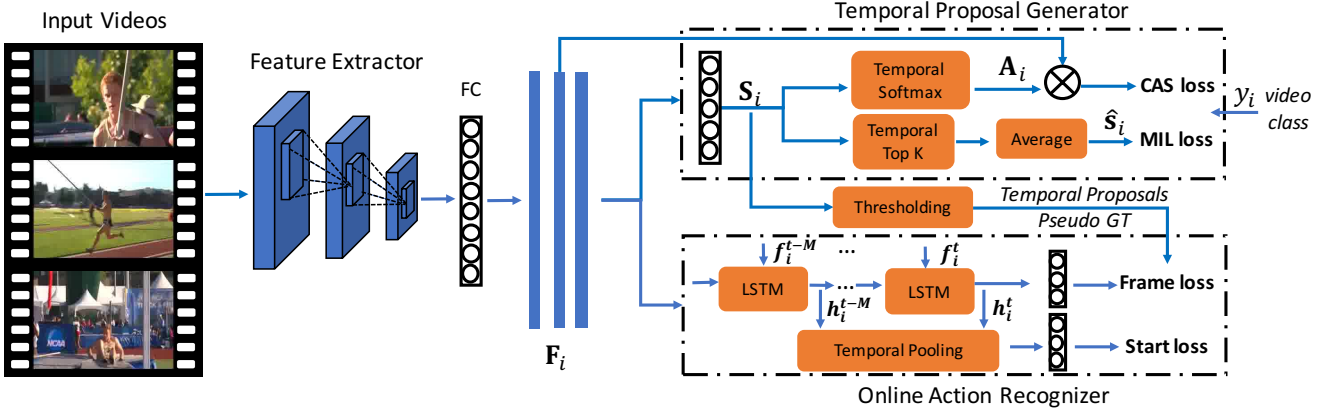


Figure 2. Illustration of the proposed WOAD in the training phase. A feature extractor is used to extract features of the input video. Frame features,  $\mathbf{F}_i$ , are then obtained by a fully connected layer (FC) activated with ReLU and serve as inputs to both Temporal Proposal Generator (TPG) and Online Action Recognizer (OAR). TPG is trained using video-class labels and its generated class-wise temporal proposals are used as pseudo ground truth of action boundaries to supervise the training of OAR. See details in Sec. 3.2 and Sec. 3.3.

inference phase. Previous approaches require annotations of temporal action boundaries for training. The proposed method can be trained using only video-class labels.

As shown in Fig. 2, our method contains two modules, i.e., the Temporal Proposal Generator (TPG) and the Online Action Recognizer (OAR). During training, TPG is supervised by video-class labels and outputs class-wise temporal proposals (Sec. 3.2). The proposals serve as pseudo ground truth of action boundaries which can be used as per-frame labels to supervise the training of OAR (Sec. 3.3). During inference, only OAR is used for online action detection.

### 3.2. Temporal Proposal Generator

There are different options to implement our Temporal Proposal Generator (TPG). In this work, we focus on utilizing Multiple Instance Learning (MIL) loss and Co-Activity Similarity (CAS) loss proposed in [22]. Next, we will revisit the definitions of these two losses

Let  $\mathbf{F}_i = [\mathbf{f}_i^1, \mathbf{f}_i^2, \dots, \mathbf{f}_i^{T_i}]^\top \in \mathbb{R}^{T_i \times D}$  indicates the features of  $\mathbf{V}_i$  just before the TPG module as shown in Fig. 2, where  $\mathbf{f}_i^t$  indicates the feature of the frame at time step  $t$ , and  $T_i$  denotes the number of frames in  $\mathbf{V}_i$ . We obtain per-frame scores,  $\mathbf{S}_i = [\mathbf{s}_i^1, \mathbf{s}_i^2, \dots, \mathbf{s}_i^{T_i}]^\top \in \mathbb{R}^{T_i \times C}$  by projecting the features to action class space and  $\mathbf{s}_i^t = [s_{i1}^t, s_{i2}^t, \dots, s_{iC}^t] \in \mathbb{R}^C$  indicates scores of frame  $t$  over  $c = [1, 2, \dots, C]$  classes. For each class  $c$ , a video-level score,  $\hat{s}_{ic}$ , is obtained by averaging over the top  $K_i$  frame scores as  $\hat{s}_{ic} = \frac{1}{K_i} \sum_{t \in \mathbb{K}_{ic}} s_{ic}^t$ , where  $\mathbb{K}_{ic}$  indicates the set of top  $K_i$  frames for class  $c$  over  $T_i$  frames,  $K_i = \lfloor \frac{T_i}{\kappa} \rfloor$  and  $\kappa$  is fixed to be 8.

**MIL loss**,  $L_{MIL}$ , is defined as the cross entropy loss between the video-class label,  $y_i$ , and the predicted video-class probability,  $p_i$ , where  $p_i$  is obtained by applying softmax in classes over  $\hat{\mathbf{s}}_i = [\hat{s}_{i1}, \hat{s}_{i2}, \dots, \hat{s}_{iC}]$ .

**CAS loss** encourages regions of videos containing sim-

ilar activities to have similar feature representations, and those containing different activities to have different representations. High- and low-attention region feature representations,  $\Psi$  and  $\Phi$ , are introduced to achieve this goal. For class  $c$ ,  $\Psi_{ic} = \mathbf{F}_i^\top \mathbf{A}_{ic}$  and  $\Phi_{ic} = \frac{1}{T_i - 1} \mathbf{F}_i^\top (1 - \mathbf{A}_{ic})$ , where  $\mathbf{A}_{ic} \in \mathbb{R}^{T_i}$  is a temporal attention vector, obtained by applying temporal softmax over frame scores,  $\mathbf{S}_i$

Intuitively,  $\Psi_{ic}$  aggregates features of regions with high probability containing the activity, while  $\Phi_{ic}$  aggregates those of regions that are unlikely involving in the activity. For class  $c$ , a positive video pair,  $\mathbf{V}_i$  and  $\mathbf{V}_j$ , is constructed if  $y_{ic} = y_{jc} = 1$ . Their pair-wise loss is calculated as

$$\frac{1}{2} \{ \max(0, d(\Psi_{ic}, \Psi_{jc}) - d(\Psi_{ic}, \Phi_{jc}) + \delta) + \max(0, d(\Psi_{ic}, \Psi_{jc}) - d(\Phi_{ic}, \Psi_{jc}) + \delta) \}, \quad (1)$$

where  $d(x, z)$  denotes cosine similarity of  $x$  and  $z$ , and  $\delta$  is a margin parameter. CAS loss,  $L_{CAS}$ , is the average loss over all positive video pairs of all classes in the training batch.

**Proposal generation** is conducted via a two-stage thresholding strategy. First, a threshold,  $\theta_{class}$ , is used to discard categories having small video-level confidence scores. Then, a second threshold,  $\theta_{score}$ , is applied on the frame scores of the remaining categories,  $s_{ic}^t$ , along the temporal axis. Taking advantage of temporal constraint of frames, nearby frames with the same category are grouped to obtain the class-wise temporal proposals and action starts are thus obtained. After that, the video-class labels are used to filter out the proposals with wrong categories.

### 3.3. Online Action Recognizer

Online Action Recognizer (OAR) sequentially takes  $\mathbf{f}_i^t$  as input and outputs per-frame action scores over classes

including background,  $\mathbf{a}_i^t \in \mathbb{R}^{(C+1)}$ , and a class-agnostic start score,  $\mathbf{st}_i^t \in \mathbb{R}^2$ , indicating the probabilities of this frame being a start point or not.

Our OAR is constructed by a LSTM with temporal pooling. The LSTM updates its hidden and cell states,  $\mathbf{h}_i^t$  and  $\mathbf{c}_i^t$ , at each time step as

$$\mathbf{h}_i^t, \mathbf{c}_i^t = LSTM(\mathbf{h}_i^{t-1}, \mathbf{c}_i^{t-1}, \mathbf{f}_i^t). \quad (2)$$

Then,  $\tilde{\mathbf{h}}_i^t$  is obtained by applying max pooling along temporal axis from  $\mathbf{h}_i^{t-M}$  up to  $\mathbf{h}_i^t$  as in Eq. 3, where  $M$  indicates the length of the temporal window.

$$\tilde{\mathbf{h}}_i^t = \max\text{pool}(\mathbf{h}_i^{t-M}, \mathbf{h}_i^{t-M+1}, \dots, \mathbf{h}_i^t). \quad (3)$$

$\mathbf{a}_i^t$  and  $\mathbf{st}_i^t$  are obtained by a linear projection followed by the softmax operation on  $\mathbf{h}_i^t$  and  $\tilde{\mathbf{h}}_i^t$ , respectively as in Eq. 4, where  $\mathbf{W}_a$  and  $\mathbf{W}_{st}$  indicate the parameters of the classifiers.

$$\begin{aligned} \mathbf{a}_i^t &= \text{softmax}(\mathbf{W}_a^\top \mathbf{h}_i^t). \\ \mathbf{st}_i^t &= \text{softmax}(\mathbf{W}_{st}^\top \tilde{\mathbf{h}}_i^t). \end{aligned} \quad (4)$$

In each training batch, we convert the proposal boundaries of each class  $c$  (obtained from TPG) to per-frame action labels,  $l_{jc}$  and binary start labels,  $\zeta_{jm}$ , where  $j = \{1, 2, \dots, \tilde{T}\}$  indicates the index of a frame,  $\tilde{T}$  is the total number of frames in the training video batch and  $m \in \{0, 1\}$  differentiates the non-start and start. We use cross entropy loss between  $l_{jc}$  and the predicted action probability,  $a_{jc}$ , to form frame loss and utilize focal loss [16] between  $\zeta_{jm}$  and the predicted start probability,  $st_{jm}$ , to construct start loss as shown in Eq. 5, where  $\gamma$  is a hyper parameter.

$$\begin{aligned} L_{OAR} &= \underbrace{-\frac{1}{\tilde{T}} \sum_{j=1}^{\tilde{T}} \sum_{c=0}^C l_{jc} \log a_{jc}}_{\text{frame loss}} \\ &\quad - \underbrace{\frac{1}{\tilde{T}} \sum_{j=1}^{\tilde{T}} \sum_{m=0}^1 \zeta_{jm} (1 - st_{jm})^\gamma \log st_{jm}}_{\text{start loss}}. \end{aligned} \quad (5)$$

### 3.4. Model Optimization and Inference

**Optimization.** Our Temporal Proposal Generator (TPG) and Online Action Recognizer (OAR) are jointly optimized by minimizing

$$L_{total} = L_{OAR} + \lambda L_{TPG}, \quad (6)$$

where  $L_{TPG} = L_{MIL} + L_{CAS}$ .  $L_{MIL}$  is computed for each videos and  $L_{CAS}$  is calculated using the positive video pairs in the training batch. Each video is segmented to non-overlapping training sequences which are used to calculate

$L_{OAR}$ . As shown in Fig. 2, proposals for OAR supervision are continuously updated. To reduce computation, we update the proposals every  $N$  training iterations.

**Inference.** For the online action detection tasks, only OAR is used during inference. Proceeding sequentially, OAR outputs  $\mathbf{a}_i^t$  and  $\mathbf{st}_i^t$  at each time step  $t$ .  $\mathbf{a}_i^t$  can be used directly as the per-frame action prediction. Following [11], scores of action starts, are obtained by  $\mathbf{as}_{i(1:C)}^t = \mathbf{a}_{i(1:C)}^t * \mathbf{st}_{i1}^t$  and  $\mathbf{as}_{i0}^t = \mathbf{a}_{i0}^t * \mathbf{st}_{i0}^t$ , where  $(1 : C)$  indicates positive classes and (0) denotes background. Then, we generate action starts following the criteria [24, 11]: (1) the predicted class  $\hat{c}_i^t = \text{argmax}(\mathbf{as}_i^t)$  is an action; (2) the maximum action score  $\mathbf{as}_{i\hat{c}_i^t}^t$  exceeds a threshold (set to be 0) and (3)  $\hat{c}_i^t \neq \hat{c}_i^{t-1}$ . As indicated,  $\mathbf{st}_i^t$  is used to boost the scores if a start is predicted at time  $t$  and suppress those otherwise.

## 4. Experiments

**Datasets.** We conduct experiments on THUMOS'14 [12], ActivityNet1.2 and ActivityNet1.3 [8]. THUMOS'14 contains 20 sport-related action classes. Following prior works, we use the validation set (200 videos) for training and evaluate on the test set (212 videos). Each video contains 15 action instances on average. ActivityNet1.2 contains 100 action classes with an average of 1.5 action instances per video. We train on the training set (4819 videos) and evaluate on validation set (2383 videos). With 200 action classes, ActivityNet1.3 is an enlarged version of ActivityNet1.2. Our model is trained with the 10k training videos and validated using its 5k validation videos. Although ActivityNet datasets are much larger, THUMOS'14 has varying video lengths and much denser temporally annotated actions which make it more challenging.

**Evaluation metrics.** Following previous works [6, 9, 28, 24, 11], frame-based average precision (F-AP) and point-based average precision (P-AP) are used as our evaluation metrics. F-AP focuses on evaluating model performance based on per-frame predictions. P-AP evaluates performance of action starts. P-AP works similarly as the bounding box based AP in the object detection task, except that P-AP uses time difference to determine whether an action start prediction is correct, while the later one uses Intersection over Union between the predicted box and the ground truth. The mean F-AP and mean P-AP are calculated by averaging F-APs and P-APs over classes, respectively.

**Baselines.** Since our TPG is implemented based on W-TALC [22], we compare with this baseline to show the advantage of our framework under the weakly supervised setting.<sup>1</sup> Then, we compare against two strongly-supervised methods, TRN [28] and StartNet [11]. TRN is a state-of-the-art

<sup>1</sup>To perform fair comparison, we evaluate on its frame-level predictions ( $\mathbf{S}_i$  in Fig. 2) during inference under the online constraint.



Methods	Sup.	mean P-AP@ Time Threshold (Seconds)										mean F-AP
		1	2	3	4	5	6	7	8	9	10	
W-TALC [22]	V	16.2	26.0	31.3	34.6	36.2	37.6	38.6	39.3	39.9	40.3	48.0
<b>WOAD</b>		<b>21.9</b>	<b>32.9</b>	<b>40.5</b>	<b>44.4</b>	<b>48.1</b>	<b>49.8</b>	<b>50.8</b>	<b>51.7</b>	<b>52.4</b>	<b>53.1</b>	<b>54.4</b>

Table 1. Comparison with our baseline under weakly supervised setting on THUMOS’14. V indicates video-level (weak) supervision.

Methods	Sup.	mean P-AP@ Time Threshold (Seconds)										mean F-AP
		1	2	3	4	5	6	7	8	9	10	
W-TALC [22]	V	5.2	8.5	10.7	12.8	14.5	15.9	17.1	18.1	19.1	20.1	53.8
<b>WOAD</b>		<b>7.9</b>	<b>11.6</b>	<b>14.3</b>	<b>16.4</b>	<b>18.8</b>	<b>20.3</b>	<b>22.2</b>	<b>23.4</b>	<b>24.7</b>	<b>25.3</b>	<b>66.7</b>

Table 2. Comparison with our baseline under weakly supervised setting on ActivityNet1.2. V indicates video-level (weak) supervision.

(SOTA) method for the online per-frame action recognition task and StartNet is the SOTA method for the task of online detection of action starts.

**Feature description.** On THUMOS’14 and ActivityNet1.2, two-stream (optical flow stream and RGB stream) I3D network [3] pre-trained on Kinetics is used as the feature extractor. Features are extracted at the chunk level. Video frames are extracted at 25 FPS and the chunk size is 16. The final features are the concatenation of the outputs of the two streams, resulting in a dimension of 2048. To perform fair comparison, our method and the baselines use the pre-extracted features provided by the authors of [22]. To avoid heavy feature extraction, we adopt the C3D features<sup>2</sup> of ActivityNet1.3 officially released by the ActivityNet Challenge.

**Implementation details.** Our method is implemented using Pytorch. The update interval of temporal proposals is set to be  $N = 100$  for THUMOS’14,  $N = 500$  for ActivityNet1.2 and 1000 for ActivityNet1.3. For OAR, the dimension of  $\mathbf{h}_i^t$  is set to be 4096 and the length of training sequence for LSTM is 64.  $M$  in temporal pooling is fixed to be 3.  $\gamma$  in Eq. 5 is set to be 2. Since starts are sparsely located in each video, we use all positive frames and randomly sample 3 times negative ones in each training batch to compute start loss.  $\lambda$  is fixed to be 0.5. Batch size of training videos is set to be 10. We use Adam [13] with weight decay  $5 \times 10^{-4}$  and set learning rate to be  $1 \times 10^{-4}$ .

**Supervision combination strategy.** When segment-level (strong) annotations exist, frame and start losses are computed using a combination of ground-truth and pseudo labels. The intuition is that the boundary annotations usually involve ambiguous decisions, so the noisy labels may serve as a type of regularization by making the label set reasonably diverse. We conduct the combination by randomly selecting 90% videos using ground-truth supervision and other videos use the noisy proposal supervision. The proposals and the combination set are updated during training.

<sup>2</sup><http://activity-net.org/challenges/2016/download.html>

## 4.1. Experimental Results

### 4.1.1 WOAD with Weak Supervision

Our main focus is weakly supervised online action detection. So, we first conduct experiments under this setting.

**Online detection of action start.** Comparisons in terms of P-AP between our approach and our baseline, W-TALC, are shown in Table 1 and 2. Our method outperforms W-TALC over all the time thresholds. Specifically, we obtain 5.7% and 2.7% higher mean P-AP when time threshold is 1 second on THUMOS’14 and ActivityNet1.2, respectively.

When only using video-level annotations, our method obtains better performance than strongly-supervised StartNet in general on THUMOS’14, ActivityNet1.2 and ActivityNet1.3 as shown in Table 3, 4 and 5.

**Online per-frame action recognition.** Comparisons between our method and W-TALC in terms of F-AP are shown in Table 1 and 2 (last column). Our method improves the baseline largely by 6.4% and 12.9% mean F-AP on THUMOS’14 and ActivityNet1.2 which demonstrate the effectiveness of our framework.

We also compare our weakly-supervised method with strongly-supervised baselines in Table 5, 6 and 8. Our weakly-supervised method achieves 54.4% mean F-AP improving the strongly-supervised TRN by 3.4% on THUMOS’14, and obtains 66.7% mean F-AP which is only 2.4% lower than TRN on ActivityNet1.2. Although using video-level labels, our method achieves comparable results to strongly-supervised TRN. THUMOS’14 is a more challenging dataset, containing  $10 \times$  action instances per video as compared to ActivityNet, so it leaves more room for our model to improve the performance. This is why our method gains much better results on THUMOS’14. For future reference, our method obtains 67.9% using weighted mIA, a newly proposed metric in [1].

### 4.1.2 WOAD with Strong Supervision

**Full strong supervision.** When using all segment-level annotations, our method largely outperforms TRN on THU-

Methods	Sup.	mean P-AP@ Time Threshold (Seconds)									
		1	2	3	4	5	6	7	8	9	10
StartNet [11]	S	<u>21.9</u>	<u>33.5</u>	39.6	42.5	46.2	46.6	47.7	48.3	48.6	49.0
<b>WOAD</b>	V	<u>21.9</u>	32.9	<u>40.5</u>	<u>44.4</u>	<u>48.1</u>	<u>49.8</u>	<u>50.8</u>	<u>51.7</u>	<u>52.4</u>	<b>53.1</b>
	S	<b>28.0</b>	<b>40.6</b>	<b>45.7</b>	<b>48.0</b>	<b>50.1</b>	<b>51.0</b>	<b>51.9</b>	<b>52.4</b>	<b>53.0</b>	<b>53.1</b>

Table 3. Comparison with strongly-supervised method for online detection of action start on THUMOS’14. V and S denote video-level (weak) and segment-level (strong) supervision, respectively. **Best** and second-best per column are highlighted.

Methods	Sup.	mean P-AP@ Time Threshold (Seconds)									
		1	2	3	4	5	6	7	8	9	10
StartNet [11]	S	7.5	11.5	14.1	<u>16.5</u>	18.4	19.7	20.9	21.8	22.9	23.6
<b>WOAD</b>	V	<u>7.9</u>	<u>11.6</u>	<u>14.3</u>	16.4	<u>18.8</u>	<u>20.3</u>	<u>22.2</u>	<u>23.4</u>	<u>24.7</u>	<u>25.3</u>
	S	<b>8.7</b>	<b>13.6</b>	<b>17.0</b>	<b>19.7</b>	<b>21.6</b>	<b>23.0</b>	<b>24.7</b>	<b>25.8</b>	<b>26.8</b>	<b>27.7</b>

Table 4. Comparison with strongly-supervised method for online detection of action start on ActivityNet1.2. V and S denote video-level (weak) and segment-level (strong) supervision, respectively. **Best** and second-best per column are highlighted.

Methods	Sup.	mean F-AP	mean P-AP@ 1
TRN [28]	S	43.8	–
StartNet [11]		–	4.9
<b>WOAD</b>	V	44.0	5.1
	S	<b>46.8</b>	<b>5.5</b>

Table 5. Comparison with strongly-supervised methods on ActivityNet1.3. V and S denote video-level (weak) and segment-level (strong) supervision.

Methods	Sup.	Param. #	Infer time	mean F-AP
TRN [28]	S	314M	2.60 ms	51.0
<b>WOAD</b>	V	110M	0.40 ms	<b>54.4</b>

Table 6. Comparison with strongly-supervised method for online per-frame action recognition on THUMOS’14. The reported times do not include the processing time of feature extraction. V and S indicate video-level (weak) and segment-level (strong) supervision.

MOS’14 (by 16.1% mean F-AP, see Table 7). On ActivityNet1.2 and ActivityNet1.3, our method achieves new state-of-the-art performance of 70.7% mean F-AP (Table 8) and 46.8% (Table 5), respectively. For online detection of action start, our method outperforms StarNet consistently for all time thresholds on both THUMOS’14 (Table 3) and ActivityNet1.2 (Table 4). Interestingly, we observe that the gap of our performance between strong- and weak-supervised settings is only 4% and 2.8% on ActivityNet1.2 and 1.3, whereas the gap is 12.7% on THUMOS’14. This may be because that the average ratio of action length over video length in ActivityNet is  $\sim 40\%$ , while the ratio is only 2% in THUMOS’14. So, our method is not as sensitive to the boundary shift of a noisy proposal in ActivityNet as in THUMOS’14.

**WOAD with Mixed Supervision.** One advantage of our method is the flexibility of taking different forms of supervi-

sion for different videos. We evaluate our model when only a portion of randomly selected videos have segment-level annotations. As shown in Table 7 and 8, the performance of our model improves when more segment-level labels are available. On ActivityNet1.2, our method achieves comparable performance to previous SOTA method when only 70% of data contains segment-level annotations.

#### 4.1.3 Model Ablation and Analysis

Our superior performance may attribute to (1) the improvements by jointly training TPG and OAR; (2) the effect of the supervision combination strategy and (3) our desirable structure. Ablation studies are conducted to analyze the effect of each component of WOAD.

**TPG and OAR joint training.** Training two modules together has following benefits: (1) the diverse pseudo labels generated in different iterations could serve as regularization during training and (2) the shared features can be potentially improved by multi-task learning. We validate its effect by removing  $L_{TPG}$  when strong labels are available. As shown in Table 9, disabling  $L_{TPG}$  ( $W/O L_{TPG}$ ) results in 5.9% and 3.5% lower mean F-AP and mean P-AP@1.

**Supervision combination.** Should we use the pseudo labels generated from TPG when strong labels are available? As shown in Table 9, using only segment-level supervision ( $W/O weak sup.$ ) results in degradation of mean F-AP by 3.2% and mean P-AP@1 by 2.6%. We observe that small amount of pseudo labels could serve as a type of regularization thus relieve overfitting. However, adding too much noise would lead to performance degradation. For example, when we use pseudo labels for 90% of videos, the mean F-AP is decreased to 58.0%.

**Effect of start point prediction.** As shown in Table 9, removing  $st_i^t$  ( $W/O st_i^t (Infer.)$ ) results in decreasing mean P-

Methods →	TRN [28]	WOAD				
Supervision →	S	V	V+10%S	V+30%S	V+50%S	S
mean F-AP →	51.0	54.4	55.0	59.3	62.6	<b>67.1</b>

Table 7. Comparison with strongly-supervised SOTA method on THUMOS’14. V+10%S means that 10% of videos have segment-level (strong) annotations and others have video-level (weak) annotations.

Methods →	TRN [28]	WOAD				
Supervision →	S	V	V+30%S	V+50%S	V+70%S	S
mean F-AP →	69.1	66.7	66.9	68.5	69.3	<b>70.7</b>

Table 8. Comparison with strongly-supervised SOTA method on ActivityNet1.2. V+30%S means that 30% of videos have segment-level (strong) annotations and others have video-level (weak) annotations.

Methods	Sup.	mean F-AP	mean P-AP@ 1
W/O RNN	V	49.0	19.7
W/O $st_i^t$ (Infer.)		54.4	20.2
W/O temp. pool		54.3	21.6
<b>WOAD</b>		<b>54.4</b>	<b>21.9</b>
W/O $L_{TPG}$	S	61.2	24.5
W/O RNN		57.5	24.6
W/O weak sup.		63.9	25.4
W/O temp. pool		65.6	26.3
<b>WOAD</b>		<b>67.1</b>	<b>28.0</b>

Table 9. Ablation study of our proposed WOAD on THUMOS’14. V and S indicate video-level (weak) and segment-level (strong) supervision, respectively.

Methods	Sup.	mean F-AP	mean P-AP@ 1
W-TALC [22]	V	48.0	16.2
<b>WOAD</b>		<b>54.4</b>	<b>21.9</b>
BaS-Net [15]	V	51.0	19.9
<b>WOAD+</b>		<b>57.8</b>	<b>22.7</b>

Table 10. Comparison of our method implemented with different modules as TPG on THUMOS’14 dataset. WOAD+ indicates our framework using BaS-Net as TPG.

AP@1 by 1.7%. As expected, start point prediction improves the accuracy of action start generation by suppressing false positives at non-start frames.

**Effect of temporal pooling.** Information of the current frame may not be the best indicator for start prediction, so we use temporal pooling to make our model more flexible to take temporal information. When it is removed, our model (W/O temp. pool in Table 9) obtains worse performances.

**Effect of RNN.** Our OAR utilizes LSTM to aggregate temporal information. To investigate the effect of the recurrent architecture, we replace the LSTM with two fully connected layers of size 4096. The performance (W/O RNN in Table 9) under both weak and strong supervised settings are much worse than our method which demonstrate the usefulness of RNN in our architecture.

**Effect of  $\lambda$ .** The hyper parameter  $\lambda$  in Eq. 6 controls the contribution of the losses from our TPG and OAR modules to the total loss.  $\lambda$  is set to be 0.5 as default. Our method is relatively robust in this hyper-parameter choice. With video-level supervision, our method achieves 54.4%, 55.0% and 54.6% mean F-AP when  $\lambda$  equals 0.5, 1.0 and 2.0, respectively. When strongly supervised, our method obtains 67.1%, 66.3% and 66.6% mean F-AP accordingly.

**WOAD with another TPG option.** Although, our TPG is based on [22], other weakly supervised techniques may also serve as TPG. We experiment with another offline weakly supervised method, BaS-Net [15], to generate action proposals. The results in Table 10 show that our method outperforms the baselines of W-TALC and BaS-Net largely by 6.4% and 6.8% mean F-AP, and by 5.7% and 2.8% mean P-AP@1, respectively. The clear gaps demonstrate the effectiveness of our design. Besides, WOAD+ performs much better than WOAD which suggests that our method achieves better results when using a more accurate TPG model.

**Model efficiency analysis.** Since our model and the baselines use the same features, we compare the inference times after feature extraction. We test all the models under the same environment with a single Tesla V100 GPU. The per-frame inference times of TRN, StartNet and our method averaging over the entire test set of THUMOS’14 are 2.60 ms, 0.56 ms and 0.40 ms respectively. The results suggest that our method is the fastest, around 6× faster than TRN. Model size is another key factor, especially for online tasks. Given similar model accuracy, smaller models are preferable, since they require less memory. Number of parameters of TRN, StartNet and our method (TPG+OAR) are 314M, 118M and 110M. Our method has the least number of parameters (3× smaller than TRN).

**Qualitative results.** In Fig. 3, we visualize our predicted action and start scores of three representative cases. As it is shown, our method performs fairly good in the case of *Long Jump*. In the second case (*Javelin Throw*), our action scores are desirable. Since the visual appearance across frames near start points are very similar, the start scores are (although still reasonable) not as good as those in the



Figure 3. Qualitative results of our weakly-supervised method. The last row of each group indicates the predicted start scores (blue bars) and the second to the last row (colored bars) indicates the predicted action scores of the ground-truth class.

first case. In the third case (*Diving*), there are two actions occurred with a very short break in between. This makes it very hard to estimate starts based on the subtle visual and motion differences. Thus, the gap of start scores between start and non-start points are relatively small.

**Our weaknesses.** Our TPG is basically an offline weakly supervised method which performs poorly in long videos especially when there are very few training videos available. The low accuracy of the generated pseudo labels may result in an unsatisfactory performance of our method in the above scenarios. We conducted experiments on TVSeries [6] dataset which contains only 20 long videos for training. With the two-stream features used in [9, 28], our method achieves 59.1% and 82.2% mean cAP in weakly and strongly supervised settings. [9] and [28] reported 79.2% and 83.7%, respectively. As suggested by the results, our method is not as competitive as it is in other datasets.

## 5. Conclusion

We propose WOAD to address online action detection using weak supervision. Previous methods rely on segment-level annotations for training which leads to significant amount of human effort and hinders the model scalability. Our proposed WOAD can be trained using only video-level labels and is largely improved when strong labels are available. Experimental results demonstrate that our method with weak supervision obtains comparable performance to the existing strongly-supervised approaches on the online action detection tasks and achieves the state-of-the-art results when strongly supervised.

**Acknowledgement.** Discussions with Peng Tang and Kathy Baxter are gratefully acknowledged. We thank Zuxuan Wu, Zeyuan Chen and Salesforce researchers for the help of improving the writing.



## References

- [1] Marcos Baptista-Ríos, Roberto J López-Sastre, Fabian Caba Heilbron, Jan C Van Gemert, F Javier Acevedo-Rodríguez, and Saturnino Maldonado-Bascón. Rethinking online action detection in untrimmed videos: A novel online evaluation protocol. *IEEE Access*, 2019.
- [2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-stream temporal action proposals. In *CVPR*, 2017.
- [3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018.
- [5] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *ICCV*, 2017.
- [6] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *ECCV*, 2016.
- [7] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. 2020.
- [8] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.
- [9] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. RED: Reinforced encoder-decoder networks for action anticipation. In *Bmvc*, 2017.
- [10] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia. TURN TAP: Temporal unit regression network for temporal action proposals. *ICCV*, 2017.
- [11] Mingfei Gao, Mingze Xu, Larry S. Davis, Richard Socher, and Caiming Xiong. Startnet: Online detection of action start in untrimmed videos. In *ICCV*, 2019.
- [12] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://crcv.ucf.edu/THUMOS14/>, 2014.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [15] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020.
- [16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [17] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, 2019.
- [18] Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *ECCV*, 2020.
- [19] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *ECCV*, 2020.
- [20] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *ICCV*, 2019.
- [21] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, 2019.
- [22] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018.
- [23] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018.
- [24] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavier Giro-i Nieto, and Shih-Fu Chang. Online action detection in untrimmed, streaming videos-modeling and evaluation. In *ECCV*, 2018.
- [25] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.
- [26] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.
- [27] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.
- [28] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S. Davis, and David J. Crandall. Temporal recurrent networks for online action detection. In *ICCV*, 2019.
- [29] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *ICLR*, 2019.
- [30] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019.
- [31] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.