SCOAD: Single-frame Click Supervision for Online Action Detection [⋆]

Na Ye[1] [0000−0001−5985−2281], Xing Zhang[1] [0000−0002−9112−4070], Dawei Yan[1] [0000−0001−5202−0255], Wei Dong[2] [0000−0003−0263−3584] [⋆⋆], and Qingsen Yan[2] [0000−0003−1010−3540] [⋆⋆]

[1] Xi'an University of Architecture and Technology
[2] Northwestern Polytechnical University

Abstract. Online action detection based on supervised learning requires heavy manual annotation, which is difficult to obtain and may be impractical in real applications. Weakly supervised online action detection (WOAD) can effectively mitigate the problem of substantial labeling costs by using video-level labels. In this paper, we revisit WOAD and propose a weakly supervised online action detection using click-level labels for training, named Single-frame Click Supervision for Online Action Detection (SCOAD). Comparatively, click-level labels can effectively improve prediction accuracy by carrying a small amount of temporal information without massively increase the difficulty and cost of annotation. Specifically, SCOAD includes two joint training modules, *i.e.*, Action Instance Miner (AIM) and Online Action Detector (OAD). To provide more guidance for training network as accuracy as possible, AIM mines pseudo-action instances under the supervision of click labels. Meanwhile, we generate video similarity instances offline by the similarity between video frames and use it to perform finer granularity filtering of error instances generated by AIM. OAD is trained jointly with AIM for online action detection by the pseudo frame-level labels converted from the filtered pseudo-action instances. We conduct extensive experiments on two benchmark datasets to demonstrate that SCOAD can effectively mine and utilize the small amount of temporal information in click-level labels. Code is available at https://github.com/zstarN70/SCOAD.git.

Keywords: Online action detection · Weakly supervised learning.

1 Introduction

Online action detection aims to report the presence of action instances in an untrimmed streaming video until the end. Unlike offline Temporal Action Lo-

[⋆] This work is supported by the Fundamental Research Funds for the Central Universities(No. D5000220444) and the Natural Science Basic Research Program of Shaanxi (2021JLM-16) and the Yulin Science and Technology Plan Project (CXY-2020-063)

[⋆⋆] Corresponding authors: (Qingsen Yan) qingsenyan@gmail.com and (Wei Dong) dongwei156outlook.com

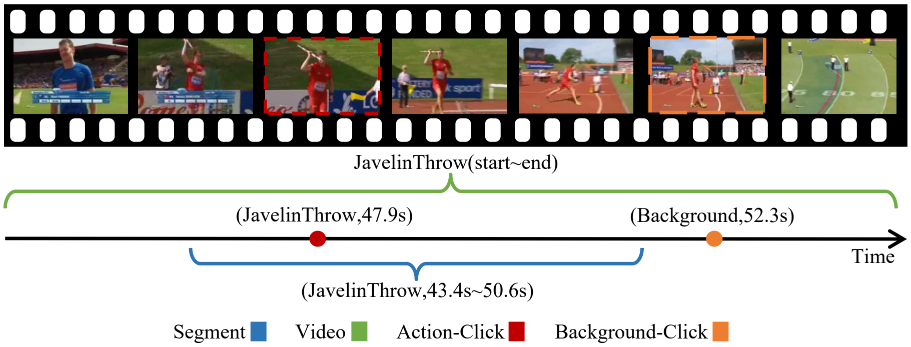


Fig. 1. Illustration of different annotation methods. (a) Segment-level annotation needs to label the action type and the precise time boundary. (b) Video-level annotation only needs to give the type of action present in the whole video. (c) Action-click labels need to give the timestamp and category corresponding to the frame where the action occurred. Similarly, Background-click happens in the background area.

calization (TAL), online action detection can only access historical frames that have been observed instead of all frames. This feature also makes it widely used in scenarios with high real-time requirements such as autonomous driving [13], anomaly detection [14] and video surveillance [21]. Therefore, this also has higher requirements for the accuracy of the algorithm and inference speed.

Current fully-supervised online action detection algorithms rely on extensive manual segment-level annotations, which are expensive to annotate. Recently, Gao *et al.* [8] considered that with the help of existing text retrieval technology, many video-level labels are relatively easy to obtain on the Internet. Therefore, WOAD was proposed to use only video-level labels, including two joint training modules, *i.e.*, Temporal Proposal Generator (TPG) and Online Action Recognizer (OAR). TPG generates pseudo temporal proposals, and OAR performs online action detection with pseudo frame-level labels generated from the pseudo temporal proposals. Although video-level labels can provide category information, almost without providing effective temporal information. Thus, WOAD tends to obtain the blurred temporal boundaries of temporal proposals generated by TPG. The current methods are limited by this problem and have poor performance for online action detection, but fewer attempts have been made to solve it. As shown in Fig. 1, click-level annotation denotes the labels of the action category and the corresponding timestamp that occurred at a random point on the video. Existing works [18,32] show that for a one-minute video, the times required to annotate video-level, click-level, and frame-level are 45s, 50s, and 300s, respectively. It demonstrates that click-level annotation without increase significantly the cost of annotation compared with video-level annotation, but click-level annotation provides more information and has capacity to improve the prediction accuracy.

With the supervision of click labels, we propose an elaborate framework, termed SCOAD, to perform online action detection through two jointly trained modules, *i.e.*, Action Instance Miner (AIM) and Online Action Detector (OAD). In our method, AIM mines the pseudo-action instances and devotes itself to understanding the video from three perspectives: the entire video, the action area, and the background area. On the one hand, AIM uses the top-k strategy to mine potential action frames on the entire video, while aggregating them by increasing the response of action-click frames. On the other hand, reducing the response of the background area can forces the network to distinguish foreground and background. These operations in AIM provides more clear boundaries of the pseudo-action instances. Furthermore, to effectively eliminate noise in the generated action instances and ensure maximum expression of network. We obtain video similarity instances based on the assumptions that actions are continuous, actions and backgrounds are separated, and actions are affine to each other. We compute the IoU value between video similarity instances and the pseudo-action instances of AIM, and reduce the noise present in the pseudo-action instance by threshold filtering. The filtered pseudo-action instances will be converted to pseudo-frame labels for OAD learning. The OAD performs online action detection under the supervision of these pseudo-frame labels and uses GRU [4] as a prediction cell for online action detection. Compared with LSTM [10], GRU has a smaller number of parameters and shorter inference time, which is suitable for scenarios with high real-time performance.

Therefore, AIM and OAD are jointly trained for online action detection under the supervision of click-label. We only use the OAD during inference, so there is no increase in inference time. We test the efficacy of Thumos14 and ActivityNet1.2 and achieve state-of-the-art performance in weakly supervised online action detection.

The contributions of this paper are as follows:

- We initially explored the application of click annotations in online action detection. In our method, we propose the SCOAD consisting of two joint training modules, which generates pseudo-action instances by AIM and performs online action detection by OAD.
- Our algorithm is more flexible. When a video is manually annotated, it can be freely switched between fully-supervision and weakly-supervision.
- Extensive experiments on two benchmarks. Compared with existing weakly-supervised methods using video-level labels, we effectively improve prediction accuracy without a significant increase in annotation cost.

2 Related Work

Online action detection Online action detection is pretty popular among various computer vision tasks [33,12,19,23,16,29,28,30,27,26]. Given an untrimmed streaming video, action instances and their classes are reported through historical and current frames that have been observed. Geest *et al.* [9] first described this problem as online action detection. Gao *et al.* proposed RED [6] to predict future sequences by taking multiple historical sequences as input. Similarly, TRN [25] improves action recognition at the current time by predicting future actions. IDU [5] considered that the input sequence may contain background and irrelevant actions, and used the traditional GRU cell [4] to judge whether to accumulate input information according to the correlation between the input sequence and the current information. Recently, Wang *et al.* noticed the existence of non-parallelism and gradient vanishing in traditional Recurrent Neural Network (RNN), that OadTR [24] was proposed to model long-term temporal dependencies based on Transformers. Focusing on category-level modeling, Yang *et al.* proposed Colar [31], an advisory paradigm mechanism. Besides, Gao *et al.* concerned that it is equally important to accurately identify the start time of an action, and proposed StartNet [7]. However, the above methods all rely on a large number of manual annotations for training, while our work uses click annotations to jointly identify the beginning of an action instance and continue to the end.

Weakly supervised online action detection Comparatively, few weakly supervised online action detection are available. As a pioneering work, Gao *et al.* first proposed WOAD [8] in online action detection using video-level annotation. In this paper, Temporal Proposal Generator (TPG) and Online Action Recognizer (OAR) are jointly trained. The former mines action instances through video-level annotations to generate pseudo-annotations for the latter. Moreover, OAR performs pooling operations in the temporal dimension for action start prediction. Although the annotations used by this method are easy to obtain, only using video-level annotations still suffers from the problem of blurred temporal boundaries.

Click-level supervision Weakly supervised temporal action detection has been extensive research [20,34,22,15,35]. Click-level supervision is an intermediate weakly supervised learning paradigm between fully supervised and weakly supervised, as Bearman *et al.* [1] first utilized point supervision for image semantic segmentation. At the video level, individual frames can also be viewed as points on the graph. SF-Net [18] is pioneering work in weakly supervised temporal action localization, exploits click-frame to mine pseudo action and background frames through supervised classification, which is further used to train a classifier. BackTAL [32] models location information and feature information through a score separation module and an affinity module, respectively.

3 Method

3.1 Overview

Given an untrimmed streaming video $\mathbf{V}_i$, the online action detection through historical and observed current frame reports of the probability of occurrence

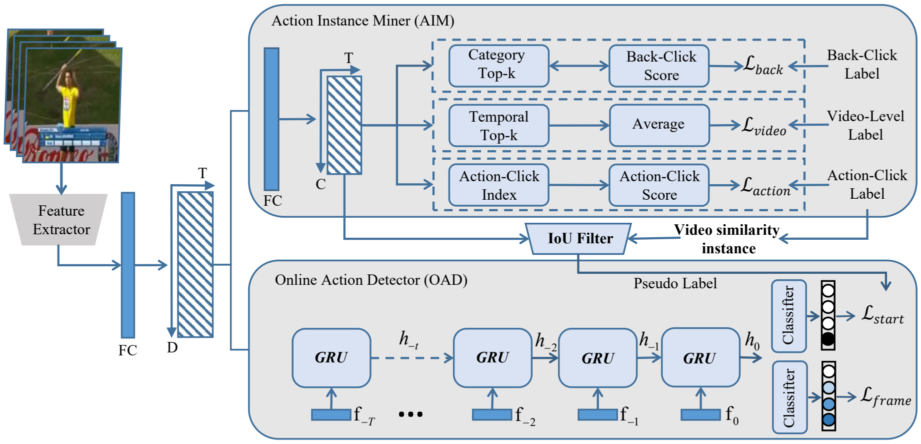


Fig. 2. The framework of click-supervised online action detection. Given a video, AIM generates pseudo-action instances under the supervision of click-level and video-level labels. After the pseudo-action instances filters itself for noise through the IoU filter, will be converted into pseudo-frame labels for OAD learning. The OAD performs online action detection under the supervision of these pseudo-frame labels.

and its category $\mathbf{y}_i = [y_0, y_1, \cdots y_c]$ for the action instance, where $y_c \in \{0, 1\}$ represents whether current frame feature $\mathbf{f}_t$ belongs to the c^{th} category.

As shown in Fig. 2, our method includes two joint training modules, *i.e.*, Action Instance Miner (AIM) and Online Action Detector (OAD). During training, AIM generates pseudo-action instances under the supervision of click labels. Then we use the IoU filter to remove noise (e.g., error in estimated results) in the pseudo-action instances, these refined labels will be converted into pseudo-frame labels for OAD learning. During inference, network is restricted from using future information and only OAD is used for online action detection.

3.2 Action Instance Miner

In this work, we attempt to mine action instances with clearer boundaries by three constraints in the Action Instance Miner (AIM). We will introduce the three loss: video-level Multiple Instance Learning [20] loss $\mathcal{L}_{\text{video}}$, action frame classification loss $\mathcal{L}_{\text{action}}$ and background score separation loss [32] $\mathcal{L}_{\text{back}}$.

Give a video $\mathbf{V}_i$, AIM takes the feature sequence $\mathbf{F}_i = [\mathbf{f}_{-T}, \ldots, \mathbf{f}_0]$ of this $\mathbf{V}_i$ as input and outputs the corresponding class score $\mathbf{s} = \{s_t\}_{t=-T}^{0}$. Its corresponding action-click labels can be expressed as $\mathbf{A}_i = \{a_t\}_{t=-T}^{0}$, in which only the frames with click annotations will be set to the corresponding c^{th}, and the rest are 0, indicating that it is uncertain whether the frame belongs to action or background. Similarly, the background labels can be denoted as $\mathbf{B}_i = \{b_t\}_{t=-T}^{0}$, $b_t \in \{0, 1\}$ indicates whether it belongs to the background.

Video-level loss $\mathcal{L}_{\textbf{video}}$ We calculate the video-level classification scores $\mathbf{s}_i^c$ in the temporal axis using top-k strategy for the c^{th} category of i^{th} video:

$$\mathbf{s}_i^c = \frac{1}{k} \max_{\substack{\mathcal{M} \subset \mathbf{s}[c,:] \\ |\mathcal{M}|=k}} \sum_{l=1}^{k} \mathcal{M}_l, \tag{1}$$

where $\mathcal{M}_l$ indicates the l^{th} element in the set $\mathcal{M}$. Finally, $\mathcal{L}_{\text{video}}$ is the cross-entropy between the predicted $\hat{\mathbf{s}}_i^c$ and video-level labels:

$$\mathcal{L}_{\text{video}} = -\frac{1}{K} \sum_{i}^{K} \mathbf{y}_i^c \log \hat{\mathbf{s}}_i^c, \tag{2}$$

where $\hat{\mathbf{s}}_i^c$ indicates the video-level classification scores after softmax normalization, $\mathbf{y}_i^c$ indicates the video includes action labels.

Action-level loss $\mathcal{L}_{\textbf{action}}$ Let us assume there are N action-click frames in $\mathbf{V}_i$, the cross-entropy loss for N action-click frames:

$$\mathcal{L}_{\text{action}} = -\frac{1}{N} \sum_{j}^{N} a_j \log \hat{\mathbf{s}_j}, \tag{3}$$

where $\hat{\mathbf{s}_j}$ indicates the action frame scores after softmax normalization.

Background-level loss $\mathcal{L}_{\textbf{back}}$ Although $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{action}}$ can employ the top-k selected positions to move closer to the region of the clicked label in the early training stage, mature models will select similar top-k positions later in training. As pointed out in BackTAL [32] study, within these regions, the model will confidently show high responses for action frames and discreetly low responses for background frames. Therefore, $\mathcal{L}_{\text{ back}}$ encourages the model to classify the responses of background and action frames distinctly.

Specifically, given a video that contains M background frames, we calculate the mean pseudo-action frame score p_{act} using the top-k strategy and the mean score p_{bg} of M background frames:

$$p_{act} = \frac{1}{k} \sum_{\forall \mathrm{b}_t=0} \mathrm{s}_t^c, \quad p_{bg} = \frac{1}{M} \sum_{\forall \mathrm{b}_t=1} \mathrm{s}_t^c. \tag{4}$$

Then, we guide $\hat{p}_{\text{act}}$ to be one while $\hat{p}_{\text{bg}}$ to be zero as follows:

$$\mathcal{L}_{\text{back}} = -\log \hat{p}_{\text{act}} - \log\left(1 - \hat{p}_{\text{bg}}\right), \tag{5}$$

where $\hat{p}_{\text{bg}}$ and $\hat{p}_{\text{act}}$ indicate the mean pseudo-action frame score and mean score of M background frames after softmax malization. Finally, we combine $\mathcal{L}_{\text{action}}$, $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{back}}$ to form $\mathcal{L}_{\text{AIM}}$:

$$\mathcal{L}_{\text{AIM}} = \mathcal{L}_{\text{action}} + \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{back}}. \tag{6}$$

Algorithm 1 Similarity Instances Mining.

```
Input:
    Video feature sequence: $\mathbf{V} = \{\mathbf{f}_t\}_{t=-T}^{0}$, $\mathbf{f_t} \in \mathbb{R}^{\mathbf{1\times N}}$
    Action-click label: $\mathbf{A} = \{c\}_{t=-T}^{0}$
Output:
    Video similarity instances: $\mathbf{S} = \{y\}_{t=-T}^{0}$
function GENERATE($\mathbf{S}$)
    for $i$ where $\mathbf{A} > 0$ do
        $k \leftarrow j \leftarrow i$
        $s \leftarrow \cos(\frac{\mathbf{V}\cdot\mathbf{f}_i}{\|\mathbf{V}\|\|\mathbf{f}_i\|})$
        $\tau = \text{mean}(s[s > 0])$
        $s[s < \tau] \leftarrow 0$
        $s[s > \tau] \leftarrow 1$
        while $s[k] \neq 0$ do
            $k \leftarrow k - 1$
        end while
        while $s[j] \neq 0$ do
            $j \leftarrow j - 1$
        end while
        $\mathbf{S}[k...j][c^{th}] \leftarrow s[k...j]$
    end for
return $\mathbf{S}$;
```

3.3 Pseudo labels generation

During the early stage of training, we use the action-click frame to calculate the similarity score with the whole video frame and generate similarity instances. The detailed process of the algorithm is summarized in Algorithm 1. During training, AIM obtains pseudo-action instances through a two-stage threshold strategy. First, categories of video-level small confidence scores are filtered using thresholds. Naturally, short instances that cannot constitute an action are filtered using threshold. Finally get the pseudo-action instances $\mathbf{I} = \{y\}_{t=-T}^{0}$, to calculate its IoU value with the video similarity instances $\theta = \text{IoU}(\mathbf{I}, \mathbf{S})$. When θ is greater than the set IoU threshold, the pseudo-action instances are converted into a pseudo-frame label for OAD learning, otherwise, the video similarity instances is converted.

3.4 Online Action Detector

Online Action Detector (OAD) takes a series of continuous feature sequence $\mathbf{F} = [\mathbf{f}_{-T}, \ldots, \mathbf{f}_0]$ as input. It outputs the corresponding action category score y_t and the probability of whether belong action start, which T is the sequence length.

In this work, OAD uses GRU as the prediction cell. The GRU updates its hidden layer h_t at each time step as:

$$h_t = \text{GRU}\left(h_{t-1}, \mathbf{f}_t\right). \tag{7}$$

Next, the fully connected layer is used to classify h_t at the current time t to obtain a_t and s_t, where a_t and s_t represent the action category score and probability of whether it belongs action start, respectively.

At the end of each training epoch, OAD obtain pseudo-action-frame labels $\mathbf{y}_{ja}^{p}$ and pseudo-action-start label y_{js}^{p} for training video from the action instances generated by AIM, where $j = \{1, 2, .., \widetilde{T}\}$ indicates the index of a frame in the training video and $\widetilde{T}$ is the total number of frames, and $y \in \{0, 1\}$ indicates the action non-start or start. Following previous work [8], we calculate the cross-entropy loss between $\mathbf{y}_{ja}^{p}$ and the action category score a_t as frame loss $\mathcal{L}_{\text{frame}}$:

$$\mathcal{L}_{\text{frame}} = -\frac{1}{\widetilde{T}} \sum_{j=1}^{\widetilde{T}} \sum_{c=0}^{C} \mathbf{y}_{ja}^{p} \log a_{jc}. \tag{8}$$

At the same time, we utilize focal loss[17] between y_{js}^{p} and predicted probability whether belong action start s_t as start loss $\mathcal{L}_{\text{start}}$:

$$\mathcal{L}_{\text{start}} = -\frac{1}{\widetilde{T}} \sum_{j=1}^{\widetilde{T}} \sum_{m=0}^{1} y_{js}^{p} \left(1 - s_{jm}\right)^{\gamma} \log s_{jm}, \tag{9}$$

where γ is a hyper parameter. Finally, we combine $\mathcal{L}_{\text{frame}}$ and $\mathcal{L}_{\text{action}}$ as $\mathcal{L}_{\text{OAD}}$:

$$\mathcal{L}_{\text{OAD}} = \mathcal{L}_{\text{frame}} + \mathcal{L}_{\text{start}}. \tag{10}$$

3.5 Trianing and inference

Training. In the early stages of training, we utilize $\mathcal{L}_{\text{AIM}}$ to optimize AIM generator pseudo-action instances. After action instances is first generated, we jointly train AIM and OAD through $\mathcal{L}_{\text{total}}$:

$$L_{\text{total}} = L_{\text{OAD}} + \lambda L_{\text{AIM}} \tag{11}$$

As shown in Fig. 2, pseudo-action instances are continuously generated by AIM. To reduce computation, we update the pseudo-action instances after every training epoch and take $Iter$ iterations as an epoch. Although we have not use the co-activity similarity loss mentioned in WOAD, that is to ensure that videos of the same category of action appear in each batch, we still split the dataset in each batch in the same way for a fair comparison.

Inference. During the inference phase, only the OAD is required for online action detection tasks. At the each time step t, OAD outputs a_t and s_t, where a_t can be used directly as the action frame prediction score. Following previous works [8,7], we obtain action start score $\widetilde{s}_t$,where $\widetilde{s}_{t(1:c)} = a_{t(1:c)} * s_{t1}$ and $\widetilde{s}_{t0} = a_{t0} * s_{t0}$ indicates c^{th} action start score and background score respectively.

Table 1. The respective performances of our method and several existing methods on THUMOS14 under different label formats are compared.

Methods	Feature	Supervison	pAP@Time Threshold(Seconds)										mAP
			1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	
W-TALC[20]$_{\text{ECCV18}}$	I3D	Video-level	16.2	26.0	31.3	34.6	36.2	37.6	38.6	39.3	39.9	40.3	48.0
WOAD[8]$_{\text{CVPR21}}$	I3D	Video-level	21.9	32.9	40.5	44.4	48.1	49.8	50.8	51.7	52.4	53.1	54.4
SCOAD	I3D	Video-level+Click-level	**24.4**	**39.2**	**44.8**	**49.0**	**50.7**	**51.6**	**52.4**	**53.0**	**53.6**	**54.0**	**61.9**
StartNet[7]$_{\text{ICCV19}}$	I3D	Frame-Level	21.9	33.5	39.6	42.5	46.2	46.6	47.7	48.3	48.6	49.0	–
TRN[25]$_{\text{ICCV19}}$	I3D	Frame-Level	–										51.0
WOAD[8]$_{\text{CVPR21}}$	I3D	Frame-Level	28.0	40.6	45.7	48.0	50.1	51.0	51.9	52.4	53.0	53.1	67.1
SCOAD	I3D	Frame-Level	**30.6**	**42.3**	**48.2**	**51.9**	**54.5**	**55.4**	**56.0**	**56.5**	**56.9**	**57.0**	**69.9**

Table 2. mAP is compared on THUMOS14 with strongly supervised and weakly supervised methods. (+x%Frame) means that x% of the videos have frame-level (strong) annotations, while the others keep their original annotations.

Methods	mAP@ Supervision(+x%Frame-Level)				
	+0%	+10%	+30%	+50%	+100%
TRN[25]$_{\text{ICCV19}}$	–				51.0
WOAD[8]$_{\text{CVPR21}}$	54.4	55.0	59.3	62.6	67.1
SCOAD	**61.9**	**63.7**	**65.2**	**66.8**	**69.9**

4 Experiments

Datasets We conduct experiments on two widely used benchmarks, THUMOS14 [11] and ActivityNet1.2 [2]. THUMOS14 includes more than 254 hours of 20 sports category videos collected from YouTube. Following previous works [8,6,5,24,31], we trained the model on the validation set (200 videos) and evaluate it on the test set (212 videos). ActivatyNet1.2 contains 9682 videos of complex human activities in 100 categories. We train on the training set (4819 videos) and evaluate on the validation set (2383 videos). The two datasets face different challenges: THUMOS14 mainly stems from the dramatic change in the duration of action instances. ActivityNet1.2 is for numerous action categories, massive intra-class changes, etc.

Evaluation metrics Following previous works [8,9,6,25,5,24], we report per-frame mean average prevision (mAP) and point-based average precision (pAP) to measure the performance of action category and action start, where mAP calculates the precision and recall for the sorting results classification scores of all frames, and then calculates the average precision of interpolation to obtain the average of the category score (AP) as the mAP. Similar to the bounding box-based AP in the object detection task, the pAP measures the accuracy of the prediction of the action start by the temporal discrepancy. We follow WOAD [8] to report pAP at these ten thresholds of [1.0 ∼ 10.0] seconds.

Table 3. The respective performances of our method and several existing methods on ActivityNet1.2 under different label formats are compared.

Methods	Feature	Supervison	pAP@Time Threshold(Seconds)										mAP
			1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	
W-TALC[20]$_{\text{ECCV18}}$	I3D	Video-level	5.2	8.5	10.7	12.8	14.5	15.9	17.1	18.1	19.1	20.1	53.8
WOAD[8]$_{\text{CVPR21}}$	I3D	Video-level	7.9	11.6	14.3	16.4	18.8	20.3	22.2	23.4	24.7	25.3	66.7
SCOAD	I3D	Video-level+Click-level	**9.2**	**12.9**	**15.9**	**18.8**	**21.1**	**22.3**	**23.7**	**24.7**	**25.6**	**26.2**	**68.7**
StartNet[7]$_{\text{ICCV19}}$	I3D	Frame-Level	7.5	11.5	14.1	16.5	18.4	19.7	20.9	21.8	22.9	23.6	–
TRN[25]$_{\text{ICCV19}}$	I3D	Frame-Level	–										69.1
WOAD[8]$_{\text{CVPR21}}$	I3D	Frame-Level	8.7	13.6	17	19.7	21.6	23	24.7	25.8	26.8	27.7	70.7
SCOAD	I3D	Frame-Level	**12.4**	**17.4**	**21.2**	**24.3**	**26.9**	**29**	**30.8**	**32.3**	**33.6**	**34.5**	**72.2**

Table 4. mAP is compared on ActivityNet1.2 with strongly supervised and weakly supervised methods. (+x%Frame) means that x% of the videos have frame-level (strong) annotations, while the others keep their original annotations.

Methods	mAP@ Supervision(+x%Frame)				
	+0%	+30%	+50%	+70%	+100%
TRN[25]$_{\text{ICCV19}}$	–				69.1
WOAD[8]$_{\text{CVPR21}}$	66.7	66.9	68.5	69.3	70.7
SCOAD	**68.7**	**70.2**	**71.1**	**71.6**	**72.2**

Baseline Since the framework of our method comes from the recent work WOAD [8], we used it as the baseline for click supervision for experiments. In our method, the LSTM cells in WOAD are replaced by GRU cells, and only the video-level Multiple Instance Learning (MIL) [20] loss is retained, and experiments are conducted to verify the effectiveness of our method.

Implementation details We conducted experiments using two-stream (RGB and optical flow) features extracted from the I3D network [3] pre-trained on the Kinetics-400 [3] dataset on THUMOS14 and ActivityNet1.2. Video frames are extracted with a frame rate of $25fps$ and chunk size is 16.

Our method is implemented on PyTorch and optimized by the Adam algorithm. We benchmark our model on an NVIDIA RTX 3090 GPU. We set batch size as 10, learning rate as 3e-4 and weight decay as 1e-4. The update proposal generation with $Iter = 100$ as an epoch on THUMOS14 and $Iter = 500$ for ActivityNet1.2 as an epoch. For OAD, we follow WOAD [8] to set hyper parameter $\gamma = 2$ in Eq. 9. The h_t dimension of the hidden layer is set to 4096, and the length of training sequence for GRU is 64. Since only few action starts exist in the video, we use all positive frames and randomly sample 3 times the number of negative frames to calculate the start loss in each training process. In addition, the click labels used in the results reported on THUMOS14 in the paper come

Table 5. Compare parameters and inference time of our method with strong and weak supervision, respectively. The reported times do not include the processing time of feature extraction.

Methods	Supervision	Param	Infer time
TRN[25]$_{\text{ICCV19}}$	Frame	314M	2.60ms
StartNet[7]$_{\text{ICCV19}}$	Frame	118M	0.56ms
WOAD[8]$_{\text{CVPR21}}$	Video	110M	0.40ms
SCOAD	Video+Click	80M	0.32ms

Table 6. Ablation study on the efficacy of each component of the AIM on the THUMOS14 dataset.

Baseline	$\mathcal{L}_{\text{action}}$	$\mathcal{L}_{\text{back}}$	IoU Filter	mAP	pAP@ 1.0
✓				49.6	17.9
✓		✓		53.7	20.7
✓	✓			56.2	22.8
✓	✓	✓		59.3	23.5
✓	✓	✓	✓	61.9	24.4

from the human click annotations provided by [18,32]. ActivityNet1.2 is the click label that we randomly generate using ground truth.

4.1 Comparison experiments

Quantitative comparisons We compare with recent state-of-the-art methods for weakly supervised online action detection and consistently obtain significant performance on THUMOS14 and ActivityNet1.2. As shown in Table 1, Our model achieves state-of-the-art performance and improves mAP on the THUMOS14 dataset from 54.4% to 61.9% compared to the WOAD. It can be seen that the more accurate click annotations can bring very intuitive performance improvements, which give a accurate clue of time information for action detection. In Table 3, compared with WOAD, the mAP of our model is improved 2.0% with click labels on ActivityNey1.2, which shows the effectiveness of the proposed method. In addition, the proposed method also can be used for action detection with frame lables. As shown in Table 3, our method outperforms WOAD with 1.5% improvement on ActivityNet1.2.

For action start prediction, our model comprehensively surpasses WOAD at every threshold in Table 1 and 3. Especially when the threshold is 1.0 second, our method pAP improves by 2.5% and 1.3% compared to WOAD on THUMOS14 and ActivityNet1.2. This shows that the classification accuracy of our method is significant. Meanwhile, compared with WOAD, our method outperforms its strongly supervised methods on both click label and frame label on THUMOS14,

Table 7. Ablation study on the efficacy of each component of the OAD on the THUMOS14 dataset.

Methods	Supervision	mAP	pAP@ 1.0
SCOAD LSTM	Video-Level+Click-Level	61.6	23.9
SCOAD Temp.pool	Video-Level+Click-Level	61.3	20.7
SCOAD	Video-Level+Click-Level	61.9	24.4
SCOAD LSTM	Frame-Level	69.6	30.4
SCOAD Temp.pool	Frame-Level	69.4	28.6
SCOAD	Frame-Level	69.9	30.6

Table 8. Randomly generated click labels on the ground truth of THUMOS14 using different random seeds.

Method	Supervision	seed	mAP	pAP@ 1.0
SCOAD	Video-Level+Click-Level	1	63.1	23.9
	Video-Level+Click-Level	10	63.1	22.6
	Video-Level+Click-Level	100	60.6	21.3
	Video-Level+Click-Level	1000	61.5	22.7

ActivityNet1.2. This phenomenon shows that choosing a suitable classifier is critical.

Effectiveness and efficiency To further illustrate the flexibility and effectiveness of our method, we also evaluate the performance using mixed annotations, as shown in Tables 2 and 4. This shows that our method can improve performance by improving the annotation accuracy. Compared to WOAD, our method without using frame-level labels approaches its performance of using 50% frame-level labels on THUMOS14 and ActivityNet1.2.

At the same time, we compare the number of parameters and computation of the model with the baseline in Table 5, and our Param and Inference time are both lower than the baseline methods and strongly supervised methods. For a fair comparison, we use the same NVIDIA Tesla V100 GPU as WOAD to calculate the average inference times on the entire THUMOS14. Benefiting from the GRU cell with fewer parameters and faster convergence, our model is the fastest, 0.08ms faster than WOAD, and has only 80M parameters.

4.2 Ablation experiments

Although we use a similar structure to WOAD, our performance outperforms WOAD. A specific reason is that our label information is more robust than WOAD. We will explore deeper reasons below through ablation experiments.

AIM of each component In Table 6, the influence of each loss in AIM on action instances generation is studied. When using the background constraint on the top-k score, we can see that both mAP and pAP grow, 4.1% and 2.8% respectively. When adding action frame constraints, mAP and pAP are improved by 6.6% and 4.9%, respectively.

Therefore, it can be inferred that although the top-k strategy can simply constrain the action classes, but a real action position is difficult to give effectively. However, although there are only a few click table labels, it can constrain the source of the top-k region. With the further restriction of back-click labels, the boundaries of actions are clearer. After that, action instances with higher confidence are further selected under the filtering of video similarity proposals. We visualize this process in Fig. 3 and it will be described in detail in Section 4.4.

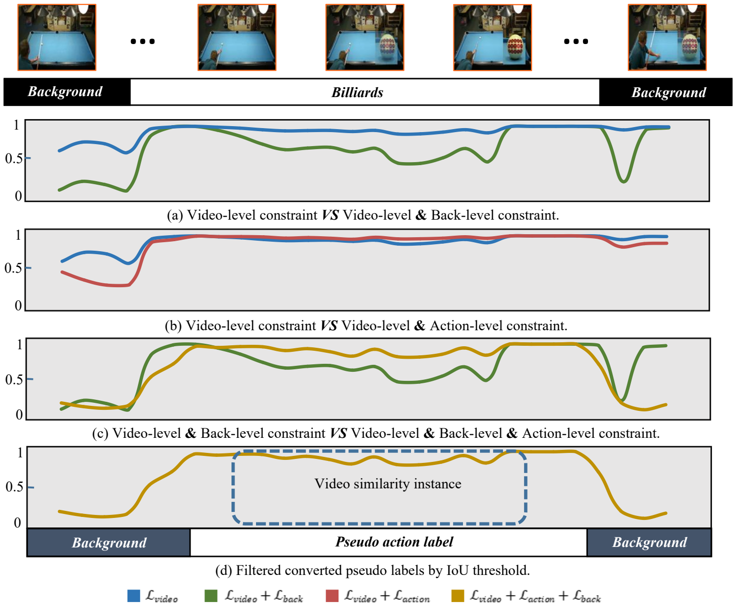
OAD of each component Compared with WOAD, we remove the max-pooling operation in the temporal dimension, directly use h_t for action start prediction in Eq. 7, and use GRU cell instead of LSTM cell. We conducted detailed experiments on the two improvements on THUMOS14, and the results are shown in Table 7. We experimented the results with different labels separately. When using click-level labels and LSTM network for prediction, mAP and pAP decreased by 0.3% and 0.5%, respectively. For action start prediction, temporal pooling will increase the convergence difficulty of the network, shown as a joint reduction of 0.6% and 3.7% in mAP and pAP. As expected, the above phenomenon also occurs when frame-level labels are used, thus validating our inferences.

4.3 Random influence of clicking labels

Since a huge variability in clicks from individual persons, elements of randomness are inevitable in our method. We devote to verifying that the performance improvement from click supervision is robust. Similar to previous work [18,32], we randomly generate several sets of click labels with different random seeds on the ground truth ofTHUMOS14. The experiment results are shown in Table 8. Each action area contains at least one action-click label, and each video contains at least one background-click label. It can be seen that the click labels generated by different random seeds bring about 2.5% and 2.6% fluctuations of mAP and pAP, respectively. But it's enough to show that click label is influential for prediction. However, how to eliminate the random factor to ensure the network converges to the same position as much as possible is still worth studying.

4.4 Qualitative results

Fig. 3 provides a quantitative analysis of our action instances generation process. As shown in Fig. 3(a), although the top-k strategy can mine action regions, it has the blurred time boundaries problem. The constraint of back-level loss can significantly respond to the background area. But it also makes the response of the action area less confident. Action-level loss constraints do not seem to fully



(a) Video-level constraint ***VS*** Video-level **&** Back-level constraint.

(b) Video-level constraint ***VS*** Video-level **&** Action-level constraint.

(c) Video-level **&** Back-level constraint ***VS*** Video-level **&** Back-level **&** Action-level constraint.

(d) Filtered converted pseudo labels by IoU threshold.

Fig. 3. Visualize the prediction scores of the AIM module under different loss constraints.

address this problem, as shown in Fig. 3(b). When the background loss and action loss are jointly employed, the predicted scores were significantly expressed in Fig. 3(c). The predicted scores showed more confident responses at action regions and background boundaries. Finally, after filtering through the IoU threshold, it is converted into a pseudo-label in Fig. 3(d).

5 Conclusion

This paper proposes a method for online action detection using click labels. Extensive experiments have demonstrated that the using click labels does not significantly increase the cost of manual annotation, but it can effectively improve the accuracy of prediction. The proposed method consists of pseudo label generation and action prediction. AIM finds potential action regions through a top-k strategy for video label and foreground label, then uses background labels to reduce the response of background regions. Thanks to these operations, the proposed method can effectively avoid the blurry boundaries. To refine the estimated results, we generates the final pseudo-labels under the filtering of similarity instances. Remarkably, exploiting the offline generated video similarity instances for action clicks brings enormous performance gains to our model. However, this video similarity proposal has prior knowledge, but it also leaves a promising direction for future research.

References

1. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: ECCV. pp. 549–565. Springer (2016)
2. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR. pp. 961–970 (2015)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
4. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: EMNLP. pp. 1724–1734 (Oct 2014)
5. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Learning to discriminate information for online action detection. In: CVPR. pp. 809–818 (2020)
6. Gao, J., Yang, Z., Nevatia, R.: Red: Reinforced encoder-decoder networks for action anticipation. In: BMVC (2017)
7. Gao, M., Xu, M., Davis, L.S., Socher, R., Xiong, C.: Startnet: Online detection of action start in untrimmed videos. In: ICCV. pp. 5542–5551 (2019)
8. Gao, M., Zhou, Y., Xu, R., Socher, R., Xiong, C.: Woad: Weakly supervised online action detection in untrimmed videos. In: CVPR. pp. 1915–1923 (2021)
9. Geest, R.D., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., Tuytelaars, T.: Online action detection. In: ECCV. pp. 269–284. Springer (2016)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation. p. 1735–1780 (nov 1997)
11. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/` (2014)
12. Kim, H.U., Koh, Y.J., Kim, C.S.: Global and local enhancement networks for paired and unpaired image enhancement. In: ECCV. pp. 339–354. Springer (2020)
13. Kim, J., Misu, T., Chen, Y.T., Tawari, A., Canny, J.: Grounding human-to-vehicle advice for self-driving vehicles. In: CVPR. pp. 10591–10599 (2019)
14. Ko, K.E., Sim, K.B.: Deep convolutional framework for abnormal behavior detection in a smart surveillance system. Engineering Applications of Artificial Intelligence **67**, 226–234 (2018)
15. Lee, P., Byun, H.: Learning action completeness from points for weakly-supervised temporal action localization. In: ICCV. pp. 13648–13657 (2021)
16. Li, J., Han, K., Wang, P., Liu, Y., Yuan, X.: Anisotropic convolutional networks for 3d semantic scene completion. In: CVPR. pp. 3351–3359 (2020)
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
18. Ma, F., Zhu, L., Yang, Y., Zha, S., Kundu, G., Feiszli, M., Shou, Z.: Sf-net: Single-frame supervision for temporal action localization. In: ECCV. pp. 420–437. Springer (2020)
19. Moran, S., Marza, P., McDonagh, S., Parisot, S., Slabaugh, G.: Deeplpf: Deep local parametric filters for image enhancement. In: CVPR. pp. 12826–12835 (2020)
20. Paul, S., Roy, S., Roy-Chowdhury, A.K.: W-talc: Weakly-supervised temporal activity localization and classification. In: ECCV. pp. 563–579 (2018)
21. Shu, T., Xie, D., Rothrock, B., Todorovic, S., Chun Zhu, S.: Joint inference of groups, events and human roles in aerial videos. In: CVPR. pp. 4576–4584 (2015)

22. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: CVPR. pp. 4325–4334 (2017)
23. Wang, P., Liu, L., Shen, C., Shen, H.T.: Order-aware convolutional pooling for video based action recognition. Pattern Recognition **91**, 357–365 (2019)
24. Wang, X., Zhang, S., Qing, Z., Shao, Y., Zuo, Z., Gao, C., Sang, N.: Oadtr: Online action detection with transformers. In: ICCV. pp. 7565–7575 (2021)
25. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: ICCV. pp. 5532–5541 (2019)
26. Yan, Q., Gong, D., Liu, Y., van den Hengel, A., Shi, J.Q.: Learning bayesian sparse networks with full experience replay for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 109–118 (2022)
27. Yan, Q., Gong, D., Shi, J.Q., van den Hengel, A., Sun, J., Zhu, Y., Zhang, Y.: High dynamic range imaging via gradient-aware context aggregation network. Pattern Recognition **122**, 108342 (2022)
28. Yan, Q., Gong, D., Shi, Q., Hengel, A.v.d., Shen, C., Reid, I., Zhang, Y.: Attention-guided network for ghost-free high dynamic range imaging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1751–1760 (2019)
29. Yan, Q., Gong, D., Zhang, Y.: Two-stream convolutional networks for blind image quality assessment. IEEE Transactions on Image Processing **28**(5), 2200–2211 (2018)
30. Yan, Q., Zhang, L., Liu, Y., Zhu, Y., Sun, J., Shi, Q., Zhang, Y.: Deep hdr imaging via a non-local network. IEEE Transactions on Image Processing **29**, 4308–4322 (2020)
31. Yang, L., Han, J., Zhang, D.: Colar: Effective and efficient online action detection by consulting exemplars. In: CVPR (2022)
32. Yang, L., Han, J., Zhao, T., Lin, T., Zhang, D., Chen, J.: Background-click supervision for temporal action localization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
33. Yu, L., Yang, Y., Huang, Z., Wang, P., Song, J., Shen, H.T.: Web video event recognition by semantic analysis from ubiquitous documents. IEEE Transactions on Image Processing **25**(12), 5689–5701 (2016)
34. Yuan, Y., Lyu, Y., Shen, X., Tsang, I., Yeung, D.Y.: Marginalized average attentional network for weakly-supervised learning. In: ICLR (2019)
35. Zhang, C., Cao, M., Yang, D., Chen, J., Zou, Y.: Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In: CVPR. pp. 16010–16019 (2021)