Adversarial Seeded Sequence Growing for Weakly-Supervised Temporal Action Localization

Chengwei Zhang[1+],Yunlu Xu[2],Zhanzhan Cheng[23*],Yi Niu[2],Shiliang Pu[2],Fei Wu[3],Futai Zou[1]
[1]Shanghai Jiao Tong University, Shanghai, China
cwzhang,zoufutai@sjtu.edu.cn
[2]Hikvision Research Institute, China
xuyunlu,chengzhanzhan,niuyi,pushiliang@hikvision.com
[3]Zhejiang University, Hangzhou, China
wufei@cs.zju.edu.cn

ABSTRACT

Temporal action localization is an important yet challenging research topic due to its various applications. Since the frame-level or segment-level annotations of untrimmed videos require amounts of labor expenditure, studies on the weakly-supervised action detection have been springing up. However, most of existing frameworks rely on Class Activation Sequence (CAS) to localize actions by minimizing the video-level classification loss, which exploits the most discriminative parts of actions but ignores the minor regions. In this paper, we propose a novel weakly-supervised framework by adversarial learning of two modules for eliminating such demerits. Specifically, the first module is designed as a well-designed Seeded Sequence Growing (SSG) Network for progressively extending seed regions (namely the highly reliable regions initialized by a CAS-based framework) to their expected boundaries. The second module is a specific classifier for mining trivial or incomplete action regions, which is trained on the shared features after erasing the seeded regions activated by SSG. In this way, a whole network composed of these two modules can be trained in an adversarial manner. The goal of the adversary is to mine features that are difficult for the action classifier. That is, erasion from SSG will force the classifier to discover minor or even new action regions on the input feature sequence, and the classifier will drive the seeds to grow, alternately. At last, we could obtain the action locations and categories from the well-trained SSG and the classifier. Extensive experiments on two public benchmarks THUMOS'14 and ActivityNet1.3 demonstrate the impressive performance of our proposed method compared with the state-of-the-arts.

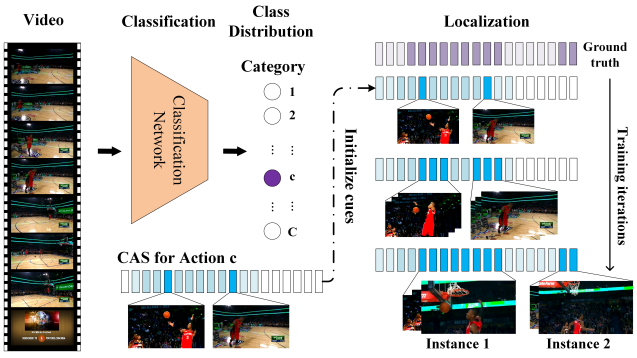


Figure 1: Seed-Grow Mechanism for Action Localization.

CCS CONCEPTS

• **Understanding multimedia content → Media interpretation**.

KEYWORDS

Temporal Action Localization, Video Understanding, Weak Supervision

ACM Reference Format:
Chengwei Zhang[1+],Yunlu Xu[2],Zhanzhan Cheng[23*],Yi Niu[2],Shiliang Pu[2],Fei Wu[3],Futai Zou[1]. 2019. Adversarial Seeded Sequence Growing for Weakly-Supervised Temporal Action Localization. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351044

1 INTRODUCTION

Temporal action localization, also called action detection, is to localize the temporal locations of actions as well as identify action categories from untrimmed videos, which is a fundamental and challenging problem in video understanding. Many existing works [1, 2, 4, 5, 14, 18, 19, 21, 27, 29, 30] make efforts to address this problem in a strong-supervised manner, where these algorithms rely on fully labeled data (*e.g.* actions annotated with precise starting and ending frames). However, untrimmed videos are usually very long, so manually annotating action locations usually seems time-consuming and expensive in real applications.

[+]Zhang partially did this work during an internship in Hikvision Research Institute.
[*]Corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351044

Above issues motivate researchers to weakly-supervised temporal actions detection only using video-level labels (i.e., action categories). Actually, weakly-supervised temporal action detection is similar to object instance detection in image with only image-level annotations (i.e., object categories). Inspired by Class Activation Map (CAM) [32] used in object detection, STPN [15] introduces the one-dimension extension of the CAM named as Temporal-CAM, also called Class Activation Sequence (CAS) in recent works [20], to locate the temporal activating regions by conducting the classification task with only video-level labels.

Following the CAS-based classification framework, [20] attempted to design an Outer-Inner-Contrastive(OIC) loss to find salient intervals, and [17] addressed the co-occurrence problem in action detection for capturing more discriminative patterns. Recently, work [28] focused on the relation learning among actions via an RNN architecture and the CAS mechanism. All the above CAS-based works are designed for weakly-supervised action localization, and achieve good performance, especially on the evaluation of low IoUs (e.g. IoU=0.1 or 0.2).

However, the CAS-based action detector usually localizes actions in untrimmed videos at the most discriminative action interval, which often appears in action response peak and results in the failure on the evaluation of high localization precision (See Figure 1). That is, such CAS-based actions detectors tend to fall into two essential issues: 1) Poor performance on the evaluation of long-duration actions due to the peak response problem caused by CAS. For example, only few discrete regions of Instance 1 are activated by CAS, which directly results in the poor results on the long-duration action detection. 2) Missing of trivial or indiscriminative actions as the case of Instance 2 missed in CAS results.

Here, we focus our attention on conquering above issues. With extensive observations, we find that though CAS tends to generate sparse activating peaks on action regions, these peaks provide important cues for mining salient parts of actions or indiscriminative actions. Therefore, an intuitive idea is that to mine more reliable action regions by referring to the estimated action cues, termed as *seeds*. Inspired by the *seed-grow* mechanism in image segmentation tasks [12], we adapt it into the temporal action localization tasks. Differently, we design the following two complementary manners of *grow* in the seeded sequences.

- We treat these activated peaks as *seeds* indicating important action cues and then extend the time durations for separated seeds to their boundaries, denoted as **the first manner of *grow***.
- Simultaneously, we erase the activated peaks from shared feature regions and further conduct a *self-adaptive classifier* for mining potential trivial or indiscriminative actions, denoted as **the second manner of *grow*.**

Above two procedures should be trained in an adversarial manner. On the one hand, erasing seeded regions of SSG will force the classifier to mine the less discriminative action regions from the feature regions. On the other hand, the classifier will also drive the seeds to grow, alternately.

In this paper, we propose a new weakly-supervised action detection framework called **Adversarial Seeded Sequence Growing (ASSG)** by adversarial learning of a **Seed Sequence Growing (SSG)** network and a **self-adaptive action classification** network. Specifically, the *SSG* is responsible for learning independent temporal heatmaps (corresponding to the action occurring probability distribution) for each action category respectively. The module takes in the most discriminative regions from CAS as the initial seeds, and progressively expands the seeded regions to neighborhood in a self-guided way. The *action classifier* devotes to exploiting the trivial missing or incomplete instances. It first erases the high-confidence regions from the SSG and thus has to find new reliable parts by the supervision of video-level class annotations. It worth noting that the module adjusts the training parameters of the shared feature maps with the SSG without any additional learning parameters, so the classifier can further promote the expanding (namely growing) of seeded regions. Consequently, these two module are trained in an adversarial manner and jointly contributes to the iteratively growing of reliable regions. Then the final results, i.e., the action locations and their categories, can be obtained from the well-trained SSG and the classifier.

The main contributions of our paper are summarized as follows:

(1) We propose an end-to-end weakly-supervised action detection approach integrating SSG network and a specific video-level classifier for mining indiscriminative action locations. To the best of our knowledge, this is the first work to introduce the *seed-grow* mechanism in temporal action detection.

(2) We train two modules in an *adversarial* manner, which not only helps grow action occurring durations and also mines trivial or indiscriminative actions.

(3) Extensive experiments demonstrate that our method achieves impressive performance on the challenging THUMOS'14 [10] and ActivityNet1.3 [7] datasets, especially on the evaluation of high IoUs which is more valuable than that in low IoUs.

2 RELATED WORK

2.1 Temporal Action Localization.

Temporal action localization aims at identifying the temporal action intervals. According to the utilized supervision information in model training, we divide existing methods into two categories: the fully-supervised based and the weakly-supervised based.

Fully-Supervised Action Detection. Most existing works generally train action detection model with frame-wisely action annotations, i.e., each action is annotated with category as well as its starting and ending position. In early time, sliding windows strategy [16] with a well-trained action classifier is the traditional solution for temporal detection. Afterwards, the proposal-based [21, 27, 30] methods were developed for effectively narrowing down the number of candidate instances. Specifically, S-CNN [21] used a multi-stage CNN for temporal action localization with extracted robust video feature representation. SSN [30] applied a watershed temporal actionness grouping algorithm (TAG) [7] for generating

action regions and then designed the structured temporal pyramid classifiers for identifying actions' categories and their localization. Inspired by object detection, SSAD [13], R-C3D [27] and TAL-Net [4] were proposed to detect one-dimensional actions by generalizing 2D spatial proposal mechanism to 1D temporal proposal form. Recently, a special boundary sensitive network (BSN) [14] attempted to locate temporal boundaries and further integrated them into action proposals. In addition, some frame-level segmentation networks [19, 29] were also developed to generate more precise action localizations by conducting the one-dimensional semantic segmentations task. However, all above works rely on frame-wisely action annotations, which are usually impractical in real applications due to the amounts of labor expenditure.

Weakly-Supervised Action Detection. Recently, action detection with only video-level labels has been studied. UntrimmedNet [24] introduced an end-to-end model for learning only single-label action categories as well as localizations, which is the first action detection approach without using frame-wise labels. STPN[15] adopted an attention module to identify a sparse subset of key action segments in a video, and fused the key segments into action regions via adaptive temporal pooling. Similarly, AutoLoc [20] directly learned the boundaries using a novel Outer-Inner-Contrastive (OIC) loss to provide the desired segment-level supervision, and W-TALC [17] introduced the Co-Activity Similarity Loss and jointly optimized it with the cross-entropy loss for weakly-supervised detecting temporal actions. The recent state-of-the-art STAR [28] focuses on the relationship learning among multiple actions, which exploits recurrent networks for assembling expected action instances into high-level feature representation, and then predicted class labels and locations for each action category step-by-step. Most of them apply the attention mechanism integrating the Class Activation Sequence(CAS) mechanism to capture the most discriminative temporal regions, while they ignore the long-duration of action occurring problem or the missing of trivial action regions. Therefore, previous methods usually falls into poor performance on the evaluation of high-IoU localization. Fortunately, recent work [31] found the contradiction between classifier and detector and thus designed a step-by-step erasion approach to train the one-by-one classifiers. This provides us the inspiration to further mine the less discriminative features from a complete video iteratively.

2.2 Weakly-Supervised Object Localization.

Weakly supervised object localization methods locate target objects using convolutional classification networks. The widely-used Class Activation Map (CAM) [32] can be used to discover the spatial distribution of discriminative parts, while they can but find very small and noncontinuous regions of the entire objects. To handle with the weakness, Hide-and-seek [23] tried to force the model to see different parts of the image and focused on multiple relevant parts of the object beyond just the most discriminative one. It is implemented by randomly masking different regions of training images in each training epoch. Erasing-based approaches [25] are proposed to mine the complementary object regions other than the former most discriminative parts, and then use the fused results as the final object localization. Furthermore, semantic segmentation methods [8, 12] treat the salient parts of object as seeds, and then iteratively expand the regions to a definite object boundary. These explorations based on image-level annotations have exemplified the weakly supervised object detection, which could provide valuable experiences to the temporal action localization.

3 METHODOLOGY

3.1 Overview

Given an untrimmed video, we traditionally trim the sequence into N segments, and encode each segment as a K-dimensional feature vector with a pre-trained two-stream video feature extractor. That is, $X = \{x_t\}_{t=1}^{N}, x_t \in \mathbb{R}^K$. Our goal is to localize all action instances in videos via an action detection model trained with only video-level annotations. Here, the annotation of each video is defined as $Y = \{y_i\}_{i=1}^{M}, y_i \in \{1, 2, \cdots, C\}$, where M is the number of categories in this single video and $\{1, 2, \cdots, C\}$ means the set of action categories.

For this purpose, we design an end-to-end action detection framework composed of two adversarial parts: 1) the Seeded Sequence Growing (SSG) module for extending action occurring durations with pre-fetched initial seeds, which is detailed in Subsection 3.2, and 2) the self-adaptive action classifier for further exploiting the missing or incomplete instances, which is detailed in Subsection 3.3. The two modules are integrated into an entire framework and trained in an adversarial manner, as shown in Figure 2.

3.2 Seeded Sequence Growing

The SSG module first uses the reliable weakly-supervised results as initial seeds to generate reliable supervision of sparse discriminative regions. Then it progressively increases the seeded temporal regions.

3.2.1 Initial Seeds. Foreground seeds can be generated from CAS-based networks [15, 20, 28]. The sparse and high reliable action regions are the peaks in the CAS activations by a relatively high threshold.

Background refers to those non-action occurring durations. Assuming that a background region is likely to appear between two action durations when it comes to a scene change, while action occurring at a time always has the consistent shot motion, we utilize the saliency detection [24] to capture shot changes as the probable *background seeds*.

Here, we formalize the initializing seeds as $\mathcal{S} = \{\mathcal{S}_c\}_{c=0}^{C}$, where $c = 0$ represents the background, and $c \in \{1, \cdots, C\}$ represents each action category respectively.

3.2.2 Backbone Network. The SSG network is learning independent temporal heatmaps for each action category. It is designed to progressively label more reliable action locations from the initial seeds $\mathcal{S}$.

Concretely, the backbone of SSG module first stacks two temporal convolutional layers (striding filters along time dimension), in which each temporal convolutional layer follows the setting as {filters=512, kernel size=1 , stride=1}. A ReLU layer follows each temporal convolution. On top of the SSG is also a temporal convolution layer but for producing the *class heatmaps* for each temporal segments $\mathcal{H} = \{\mathcal{H}_{c,t} | c \in \{0, \cdots, C\}, t \in \{1, \cdots, N\}\}$, in which

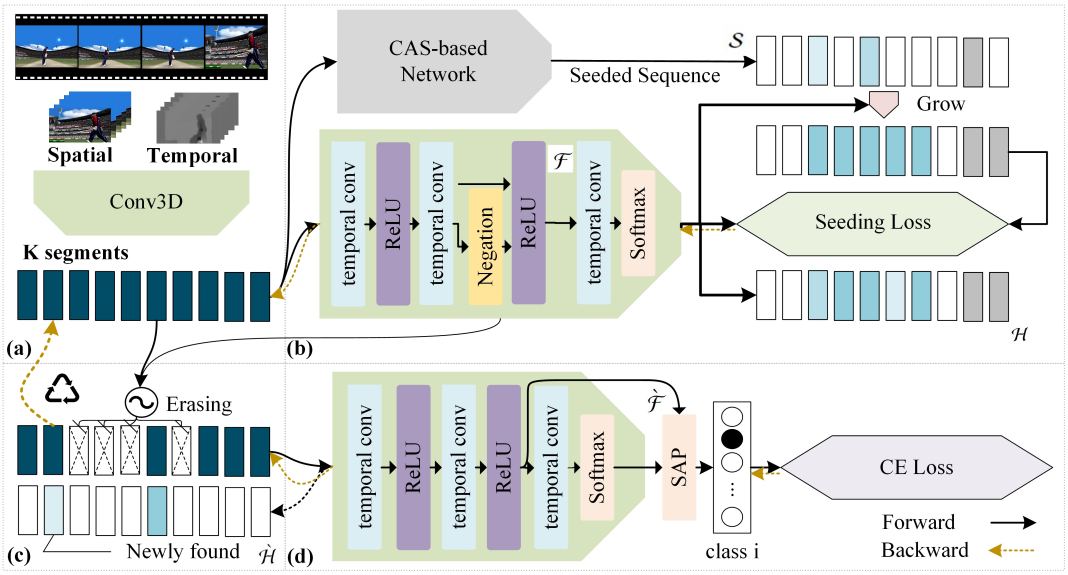


Figure 2: The proposed ASSG architecture. (a) Encoded segmental features from video inputs. (b) The SSG module, with CAS-based results as initial seeds, iteratively extending the high reliable action or background regions using growing rules, which learns the category of reliable input segments by a seeding loss. (c) Erasion of the seeding regions from the shared feature map from SSG. (d) An action classifier, which uses self-adapted pooling (SAP) for feature aggregation into the final class confidence with the cross-entropy loss.

$\mathcal{H}_{c,t}$ means the class c probability distribution of the t-th segment in the video.

3.2.3 Growing Strategy. Inspired by the growing strategy of dynamic supervision [8], we propose a one-dimension growing policy to dynamically enlarge the seeded sequences on each independent class heatmap $\mathcal{H}_{c,t}$. Once initialized, the current action regions are grown by expanding those seeds $\mathcal{S}$ to neighboring unlabeled locations $\mathcal{N}(\mathcal{S})$ via the growing criterion $\mathcal{G}$:

$$\mathcal{G}(\mathcal{H}_{c,t}, \mathcal{S}_c, \theta_g^c) = \begin{cases} True, & l \in \mathcal{N}(\mathcal{S}_c),\ \mathcal{H}_{c,t} \geq \theta_g^c \\ & \text{and } c = \arg\max_{c'} \mathcal{H}_{c',t} \\ False, & otherwise. \end{cases}, \quad (1)$$

where θ_g is the pre-set *growing threshold* value respectively for each action class and the background. Here, we use a simple definition of $\mathcal{N}(\mathcal{S})$ to be the set of locations next to each seed in $\mathcal{S}$. Considering this criterion, if $\mathcal{G}$ is true, we label the category of the t-th segment with class c as newly added supervision regions. Iteratively, we motivate the activations of both the action occurring and the background durations on heatmaps alternately with the grown supervision.

In practice, since the *co-occurrence* locations can not be assigned to two different classes when conducting the *seed-grow* mechanism in original temporal segmentation framework, we generate separated seeds for each action class (including the background) and expand the seeded regions respectively. That is, the SSG predicts individual action occurring regions one-by-one with growing policy for each class.

3.3 Self-Adaptive Action Classifier

This module is designed to mine the relatively long or trivial actions, which shares the feature map with SSG. It first erases the most discriminative part dynamically activated by SSG, and then predicts the action class by directly *aggregating* the output of shared feature maps into classification confidence scores. In this way, the classifier adaptively updates the shared parameters supervised by video class annotations without adding extra parameters.

To be specific, in the first step, we *extract* the foreground feature maps from the entire maps $\mathcal{F}$ in SSG, and then *erase* the highly activated regions to generate the remaining feature maps $\grave{\mathcal{F}}$. As $\mathcal{F}$ contains mixed activations of the foreground and the background, we need to draw only the foreground features for classification. Therefore, a pair of opposite ReLU activations is designed to generate $\mathcal{F}$, which forces the foreground or background seeds to grow in the positive or negative activation parts respectively. Naturally, the classifier can obtain the foreground features by a ReLU layer. The erasion is simply implemented by thresholding on the activation values.

The next step is the *aggregation* of feature maps. The common *aggregating* approach like global max-poling (GMP) [16] or global average-pooling (GAP) [32] is not suitable for this task, since the former ignores too many less discriminative regions and in the later case, the global feature will be overwhelmed by the large-scale occupied background segments. For the purpose of inspiring full potential of the classifier, we design a **Self-Adaptive Pooling (SAP)** approach for straightforward video class prediction. The SAP is to re-balance the weights of segments with an off-the-shelf attention weights $A_{c,t}(X)$ for class c at temporal location t, which

could be achieved from the learned feature maps shared with the SSG as

$$SAP(X) = \sum_{t=1}^{N} A_{*,t}(X) \cdot \grave{\mathcal{H}}_{*,t}. \tag{2}$$

Here, $\cdot$ is the dot product operation of two scalars. Similarly to $\mathcal{H}$, $\grave{\mathcal{H}}$ here is the activation distribution after erasion. The self-adaptive weighted aggregation of $\grave{\mathcal{H}}$ over the entire N temporal segments results in the entire video-level class distribution.

In the equation, the attention weights $A_{c,t}(X)$ can be formulated by

$$A_{c,t}(X) = \frac{e^{(\sum_{i=1}^{|f_{c,t}(X)|} f_{c,t}^{i}(X))}}{\sum_{t=1}^{N} e^{(\sum_{i=1}^{|f_{c,t}(X)|} f_{c,t}^{i}(X))}}, \tag{3}$$

where $f(\cdot)$ represents the mapping functions for the foreground features $\grave{\mathcal{F}}$ from the network inputs. $|f_{c,t}(X)|$ represents the feature dimensions at each location on $\mathcal{F}$.

Note that, assuming the highly activated regions are likely to be the interested action occurrences, $A(X)$ is the self-adaptive weight directly computed from the feature map without any extra explicit attention modules, .

3.4 Training of ASSG

The two modules (i.e.,the SSG and the action classifier) are integrated into a whole network and trained in an adversarial manner.

3.4.1 Seeded Sequence Growing Loss. T_c is a set of temporal locations that are identified as action class c. Here, the seeding loss L_{seed} is defined as:

$$L_{seed} = -\frac{1}{\sum_{c\in[0,C]} |T_c|} \sum_{c\in[0,C]} \sum_{t\in T_c} log\mathcal{H}_{c,t} \tag{4}$$

The SSG learns the parameter by optimizing the seeding loss function L_{seed}, which encourages the networks to match reliable localization cues T_c, including the foreground ($c \in [1, C]$) and the background ($c = 0$), but is agnostic about the rest of the locations.

3.4.2 Action Classification Loss. The *classification loss* is defined as the cross-entropy loss over multiple categories by

$$L_{class} = -\sum_{c=1}^{C} \hat{y}_c \log SAP(\hat{y}_c|X), \tag{5}$$

where $\hat{y}_c$ represents the ground truth of the action category.

3.4.3 End-to-end Training. The whole network is trained in an adversarial manner. By erasing the seeded regions activated by SSG, the classifier branch poses a more difficult task to discover minor or even new action regions. Alternately, the classifier will also boost the seeds growing and generate more reliable regions. The training procedures are illustrated in Algorithm 1.

3.5 Location Prediction

We use the predicted heatmap by the SSG module to generate temporal action proposals. As $\mathcal{H}_{c,t}$ denoted in Subsection 3.2.2 indicates the probability in the class heatmaps. Here, we fuse the separately trained two-stream network predictions. For each class c in the corresponding heatmap, each proposal $[N_{start}, N_{end}]$ is assigned to a score by:

$$\sum_{t=N_{start}}^{N_{end}} \frac{[\lambda \cdot \mathcal{H}_{c,t}^{RGB} + (1-\lambda) \cdot \mathcal{H}_{c,t}^{flow}]}{N_{end} - N_{start} + 1} \tag{6}$$

in which we fuse the probability values of RGB and optical flow streams by the modality ratio λ. For final detection, we perform non-maximum suppression (NMS) among temporal proposals of each class by removing highly overlapped ones.

Algorithm 1 Framework of **ASSG**.

Input: Training data, $\mathcal{V} = \{\mathcal{X}, y_c\}$; growing threshold, θ_g; adversarial threshold θ_a; sequence length N; total categories C; initial seeds $\mathcal{S}$
Output: Enhanced CAS heatmaps $\mathcal{H}$; prediction label c;

```
Initialize H;
Initialize the shared deep feature map F;
while training not converge do
    for t = 0 → N − 1 do
        for c = 0 → C do
            if G(H_{c,t}(X), S, θ_g^c) then
                the location at t-th segment is labeled as c;
            else
                the location at t-th segment keeps unlabeled state;
            end if
        end for
    end for
    Update H with the seeding loss L_seed.
    Obtain high reliable regions: U_c = {t|H_{c,t} > θ_a };
    Erase clips at temporal location in sets {U_c}_{c=1}^{C} for each category from F to obtain F̀;
    Calculate SAP(X);
    Update F̀,F (sharing parameters) with the cross-entropy loss L_class.
    Compute updated H;
end while
```

4 EXPERIMENTS

4.1 Datasets and Evaluation

Datasets. THUMOS'14 [10] is a popular dataset for action localization task, which consists of 20 action classes, and only its validation and testing set contains temporal annotations. Since the training set contains no temporal annotations, the fully-supervised algorithms use the 212 untrimmed videos in validation set to train the network and the 200 testing videos to evaluate the algorithm. To facilitate comparisons, we follow this conventional protocol but without using the temporal annotations for training. **ActivityNet v1.3 [7]** covers 200 activity classes. We also use the 10,024 training set videos without temporal annotations for training and 4926 validation videos for validating. In this section, we report our results on both THUMOS'14 and ActivityNet v1.3.

Evalution Metrics. In action localization task, mean Average Precision (mAP) is used as evaluation metrics, where Average Precision (AP) is calculated on each action class when the prediction is classified correctly and its temporal overlap Intersection over Union (IoU) with the ground truth segments exceeding the evaluation threshold. The ablation study are performed on the THUMOS'14 dataset and evaluated with average mAP (Ave-mAP) by calculating the multiple overlap IoU with thresholds varying from 0.1 to 0.5. The overall performance compared with the state-of-the-arts is evaluated with average mAP from 0.1 to 0.5 on THUMOS'14 and 0.5 to 0.95 on Activitynet v1.3.

4.2 Implementation Details

We implement our algorithm using Caffe [9]. For comparison, we employ the common procedures described in [15, 28] to uniformly sample 400 segments from each video. For extracting visual features, then we use the two-stream I3D network described in [3] pre-trained on Kinetics dataset [11]. For the CAS-based network, we realize the ST-GradCAM with the pre-defined parameters described in [28] as our default setting. Note that we train the ST-GradCAM without a specific sub-module for repetition alignment, which needs additional annotations of action frequency. Our proposed network is trained by Adam optimizer with initial learning rate 10^{-4} on both streams. For the growing threshold θ_g, we set both foreground and background threshold as 0.99. Besides, the earsing threshold θ_a is set to 0.4 on both datasets. In location prediction, we empirically set λ to 0.3.

4.3 Ablation Study

We conduct ablation study on two pre-defined thresholds, effect of feature aggregation methods and different modules. Qualitative evaluation is also provided in this section.

4.3.1 Thresholds. We pre-define two thresholds in our ASSG framework: θ_g for deciding to label the segment or not in the seeded sequence growing rule, and θ_a for erasing the high activated regions from SSG in the classification branch, respectively.

For the growing threshold θ_g, we could set different thresholds for different classes and background respectively. For convenience, we set the same threshold number for all the actions as foreground threshold θ_{gf} and another background threshold θ_{gb}. Then we fix the $\theta_{gb} = 0.99$ and vary the θ_{gf} from 0.8 to 0.95 with a step size 0.05, vice verse. As shown in Figure 4(a), the choice of different threshold values has small influence on the final results, which is convenient and robust for network training.

For the adversarial threshold θ_a, it strikes a balance between the adversarial training of two modules. We set the threshold from 0.3 to 0.7 and observe that the localization performance is boosted when the threshold θ_a = 0.4 as shown in Figure 4(b). Larger value or smaller would introduce more background noises and thus slightly influence the minor region mining.

4.3.2 Feature Aggregation. In the self-adaptive action classifier module, we aggregate the shared features directly into classification scores without any additional learning parameters. Instead of the common GMP and GAP approach, we introduce an aggregation method called SAP in Subsection 3.3. Here we discuss the different aggregation types used in our classifier in Table 1.

We conclude that the low performance of GAP is mainly because the foreground features are overwhelmed by the background segments, while GMP ignores too many regions. Our proposed SAP achieves best performance at all IoUs compared with the existing GMP and GAP, which verifies the effectiveness of our well-designed self-adaptive weighted aggregation method SAP.

Table 1: Evaluation of different aggregation methods in terms of mAP@IoU on THUMOS'14.

Methods	0.1	0.2	0.3	0.4	0.5
GMP	52.8	46.4	37.6	29.0	18.4
GAP	50.6	45.1	36.5	27.9	17.5
SAP	**60.1**	**54.6**	**45.1**	**34.3**	**22.4**

4.3.3 Architecture and Modules. We study the effects of different modules in the entire framework. The CAS, CAS w/ SSG (short in SSG), and the CAS w/ SSG w/ classifier (short in ASSG) results are shown in Table 2. The SSG module boosts the evaluation of the top reported CAS average mAP [28] from 24.4% to 34.2%. And the additional classification branch for adversarial training jointly denoted as ASSG further increases the mAP value of SSG by 9.3%, definitely a large margin. Each of the two modules plays an important role in improving the detection results.

Table 2: Evaluation of different modules in ASSG on THUMOS'14.

Reported CAS [28]	√	√	√
SSG (CAS w/ SSG)		√	√
ASSG (CAS w/ SSG w/ classifier)			√
Ave-mAP(%)	24.4	34.2	**43.5**

4.3.4 Qualitative Evaluation. Visualization results are shown in Figure 3. We randomly select a test video in THUMOS'14 dataset (including the action *Hammer Throw*) for the trained proposed ASSG network initialized with the reimplemented CAS baselines [28]. For further demonstrating the performance of ASSG, we qualitatively analyze the effective of the *growing* mechanism in different parts of the entire network.

- The *seeds* are thresholded results from reimplemented top CAS-based network [28], which effectively detects the most discriminative parts in the videos by a recognition network and fails in the evaluation of high quality detection shown as the missing regional or entire action durations highlighted in *yellow* on top line. For instance, in the first region denoted in yellow rectangles, only the start and end point are activated and in the later two yellow-filled parts, the entire action instances are missing.

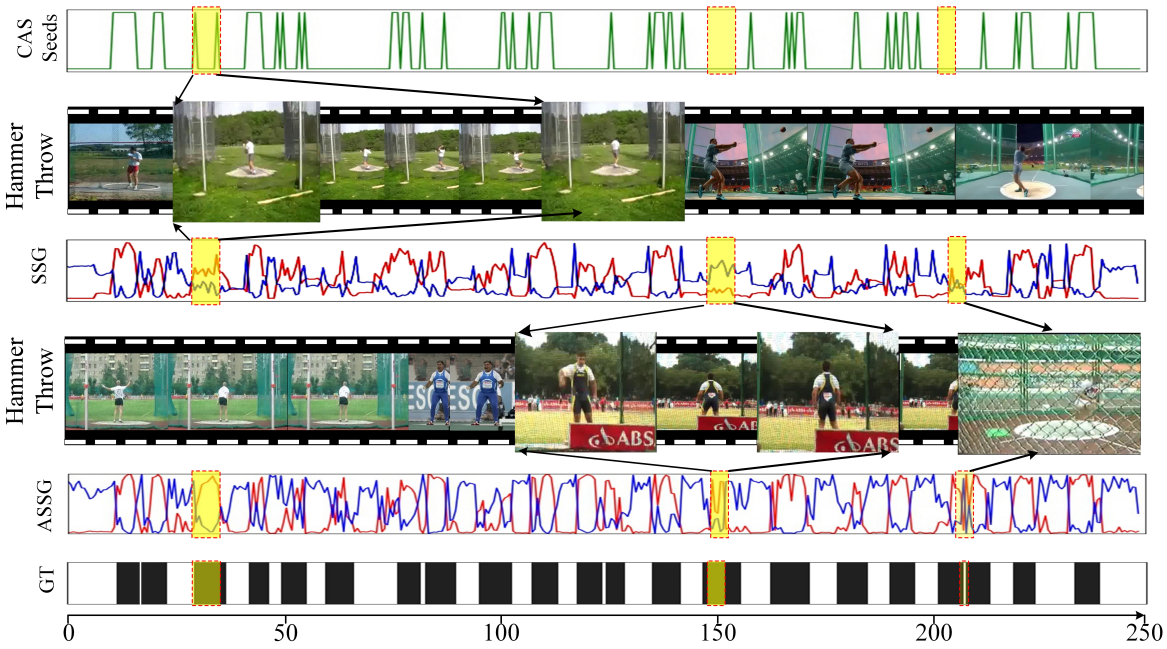
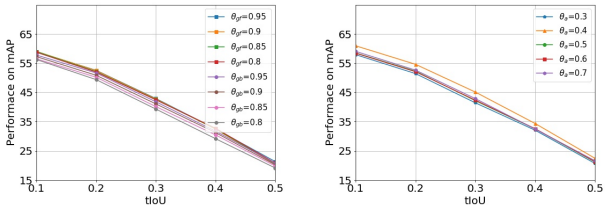


Figure 3: Visualization of action localization by ASSG network. Temporal confidence distribution (the predicted heatmap in SSG) of the action class *Hanmmer Throw* is denoted in red curve and the prediction of the *background* is in blue curve. The *yellow* locations are selected samples of enhanced detection compared with the common CAS results.



(a) The growing threshold θ_g (b) The erasing threshold θ_a

Figure 4: Performance with different thresholds values. Figure (a) shows mAP evaluation of different θ_g choices, where θ_{gf} and θ_{gb} means the foreground and background threshold respectively. Figure (b) shows mAP evaluation of different θ_a choices.

- We can obviously find the SSG detections expand to some less discriminative parts, which could lead to better detection proposals than the direct CAS result. The improvement verifies the effectiveness of seed-grow mechanism introduced into the SSG for temporal action detection.
- Finally, the entire proposed network *ASSG* leads to a more satisfying localization results as the action and complementary background regions interconnect each other tightly. When compared with the single SSG module, we also find that the independent prediction of each foreground class and the background respectively in ASSG holds more confidence (i.e. the prediction probabilities are much higher than the single SSG). We attribute the advantage to the adversarial training of the two modules, which makes it more difficult for the classifier to identify the video class and thus effectively motivates it to mine more minor regions and to expand the small seeded region to its precise boundaries.

4.4 State-of-the-Art Comparisons

We compare our model with the state-of-the-art weakly-supervised and fully-supervised methods on THUMOS'14 and ActivityNet1.3 benchmarks. Table 3 and Table 4 summarize the results.

Results on THUMOS'14. The comparison results between the proposed ASSG and other state-of-the-art models are shown in Table 3. As ASSG is a framework enhancing the CAS detections, its seeds can be initialized by various CAS-based models. In this perspective, we conduct ASSG based on two typical structure of CAS-based networks, respectively with CAS results from STPN [15] as initial seeds (denoted as *STPN-CAS w/ ASSG*) and CAS results from a constrained STAR [28] (excluding the specific sub-module with additional action frequency annotation) as initial seeds (denoted as *STAR-CAS w/ ASSG*). For a fair comparison, we also follow their prediction operations to fuse attention weights with the CAS by generating proposals separately.

We find that STPN-CAS w/ ASSG improves the performance by a large margin compared to the original STPN result, similarly in STAR-CAS w/ ASSG compared with STAR. The STAR-CAS w/ ASSG outperforms all other *weakly-supervised* methods with multiple overlap IoU thresholds varied from 0.3 to 0.6. For instance,

when the IoU threshold used in evaluation is set to 0.5, ASSG network improves the state-of-the-art results from 23.0% to 25.4%. It is noting that at IoU threshold of 0.1 and 0.2, our ASSG still achieves impressive performance, which surpasses all other methods except for STAR [28]. While STAR used additional action frequency annotations (different from all existed weakly-supervised works), and it also reported results without frequency annotations, i.e., avg-mAP ranging from 0.1 to 0.5 is 44.0%, which falls behind our ASSG 47.9% by a 3.9%. In specific, we attribute the effectiveness of our approach above CAS-based results [15, 17, 20, 28] (especially in higher IoU thresholds) to the ability of ASSG to mine less discriminative action regions, which results in more precise boundaries and completeness of detection.

We also compare the results with the *fully-supervised* methods. The performance of our weakly-supervised model with only video-level class annotations in training, even achieves comparable results with the state-of-the-art strong-supervised ones, by only 0.4% behind the TALNet [4] at IoU threshold of 0.1 and 3.1% behind the BSN [14] at IoU threshold of 0.3.

Results on ActivityNet1.3. Table 4 shows the results on ActivityNet1.3 dataset. Note that, the dataset characteristics differ largely from those in THUMOS'14 that many videos in ActivityNet1.3 are including relatively long action durations per instance. Therefore, THUMOS'14 is a better dataset for evaluating most action localization methods, which is also claimed in [4]. Since our designed framework is to expand the action regions from the initial small and sparse reliable regions, connecting all the discrete parts into a unified long-duration action poses a great challenge.

Results show that ASSG gets better overall performance than all the existing weakly-supervised results by increasing the mAP at IoU threshold of 0.5 and 0.75 from 31.1% to 32.3% and 18.8% to 20.1% respectively. We do not care about the mAP at IoU threshold of 0.95 since the precision is relatively low, which makes little sense in current situations. Although the performance gain is smaller compared to that in THUMOS'14, the improvement performance also verifies the common effectiveness on both datasets.

Table 3: Comparison with state-of-the-arts on THUMOS'14.

Method	Label	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Richard et al. [18]	strong	39.7	35.7	30.0	23.2	15.2	–	–
S-CNN [21]	strong	47.7	43.5	36.3	28.7	19.0	10.3	5.3
CDC [19]	strong	–	–	40.1	29.4	23.3	13.1	7.9
Gao et al. [6]	strong	54.0	50.9	44.1	34.9	25.6	19.1	9.9
Xu et al. [27]	strong	54.5	51.5	44.8	35.6	28.9	–	–
SSN [30]	strong	**66.0**	**59.4**	51.9	41.0	29.8	19.6	10.7
SSAD [13]	strong	50.1	47.8	43.0	35.0	24.6	–	–
TPC [29]	strong	–	–	44.1	37.1	28.2	20.6	12.7
TALNet [4]	strong	59.8	57.1	53.2	**48.5**	**42.8**	–	–
Alwasssel et al.[1]	strong	49.6	44.3	38.1	28.4	19.8	–	–
BSN [14]	strong	–	–	**53.5**	45.0	36.9	**33.8**	**20.8**
UntrimmedNet [24]	weak	44.4	37.7	28.2	21.1	13.7	–	–
Hide-and-Seek [23]	weak	36.4	27.8	19.5	12.7	6.8	–	–
Zhong et al. [31]	weak	45.8	39.0	31.1	22.5	15.9	–	–
AutoLoc [20]	weak	–	–	35.8	29.0	21.2	**13.4**	5.8
W-TALC [17]	weak	55.2	49.6	40.1	31.1	22.8	–	**7.6**
STPN [15]	weak	52.0	44.7	35.5	25.8	16.9	9.9	4.3
STAR [28]	weak	**68.8**	**60.0**	**48.7**	**34.7**	**23.0**	–	–
STPN-CAS w/ ASSG	weak	55.6	49.5	41.1	31.5	20.9	13.7	5.9
STAR-CAS w/ ASSG	weak	65.6	59.4	**50.4**	**38.7**	**25.4**	**15.0**	6.6

Table 4: Comparison with state-of-the-arts on ActivityNet1.3.

Method	Label	0.5	0.75	0.95
Singh et al. [22]	strong	34.5	–	–
CDC [19]	strong	45.3	26.0	0.2
SSN [26]	strong	39.1	23.5	5.5
SSAD [13]	strong	49.0	32.9	7.9
Chao et al. [4]	strong	38.2	18.3	1.3
BSN [14]	strong	**52.5**	**33.5**	**8.9**
STPN [15]	weak	29.3	16.9	2.6
STAR [28]	weak	**31.1**	**18.8**	**4.7**
STAR-CAS w/ ASSG	weak	**32.3**	**20.1**	4.0

5 CONCLUSION

By observing the weakness of CAS-based approach that only the most discriminative parts can be detected, we extend the *seed-grow* mechanism to our weakly-supervised temporal action detection. We design a framework of two modules, an SSG and an action classifier respectively, which jointly help small and sparse action occurring durations grow. The two modules are trained in an adversarial manner. The operation of erasing seeded regions forces the classifier to handle with a more difficult task by focusing on less discriminative regions. Alternately, the classifier drives the seeds to grow. Extensive experiments demonstrate that our ASSG achieves superior performance on the challenging THUMOS'14 above all other weakly-supervised methods, especially on the evaluation of high IoUs, and has impressive results on the ActivityNet1.3 datasets as well.

6 ACKNOWLEDGEMENT

Chengwei Zhang and Futai Zou were partially supported by National Key Research and Development Program of China (No.2017 YFB0802300), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (No. U1509219), and National Key Research and Development Program of China (No. 2018YFB 0803500). THANKS to all members in DAVAR lab.

REFERENCES

[1] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. 2018. Action Search: Spotting Actions in Videos and Its Application to Temporal Action Localization. In *ECCV*. 251–266.
[2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. 2017. SST: Single-Stream Temporal Action Proposals. In *CVPR*. 6373–6382.
[3] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*. 4724–4733.
[4] Yu Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*. 2933–2942.
[5] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S. Davis, and Yan Qiu Chen. 2017. Temporal Context Network for Activity Localization in Videos. In *ICCV*. 5727–5736.
[6] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. 2017. Cascaded Boundary Regression for Temporal Action Detection. *CoRR* abs/1705.01180 (2017).

[7] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*. 961–970.

[8] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. 2018. Weakly-Supervised Semantic Segmentation Network with Deep Seeded Region Growing. In *CVPR*. 7014–7023.

[9] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *ACM MM*, 675–678.

[10] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/.

[11] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017).

[12] Alexander Kolesnikov and Christoph H. Lampert. 2016. Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation. In *ECCV*.

[13] Tianwei Lin, Xu Zhao, and Zheng Shou. 2017. Single Shot Temporal Action Detection. In *ACM MM*. 988–996.

[14] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *ECCV*. 3–19.

[15] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. 2018. Weakly Supervised Action Localization by Sparse Temporal Pooling Network. In *CVPR*. 6752–6761.

[16] Dan Oneata, Jakob J Verbeek, and Cordelia Schmid. 2014. Efficient Action Localization with Approximately Normalized Fisher Vectors. In *CVPR*. 2545–2552.

[17] Sujoy Paul, Sourya Roy, Amit K Roy Chowdhury, and Amit K. 2018. W-TALC: Weakly-supervised Temporal Activity Localization and Classification. In *ECCV*. 563–579.

[18] Alexander Richard and Juergen Gall. 2016. Temporal Action Detection Using a Statistical Language Model. In *CVPR*. 3131–3140.

[19] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. 2017. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*. 5734–5743.

[20] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. 2018. AutoLoc: Weakly-supervised Temporal Action Localization in Untrimmed Videos. In *ECCV*. 154–171.

[21] Zheng Shou, Dongang Wang, and Shih Fu Chang. 2016. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *CVPR*. 1049–1058.

[22] Gurkirt Singh and Fabio Cuzzolin. 2016. Untrimmed Video Classification for Activity Detection: submission to ActivityNet Challenge. *CoRR* abs/1607.01979 (2016).

[23] Krishna Kumar Singh and Jae Lee Yong. 2017. Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization. In *ICCV*. 3544–3553.

[24] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. 2017. UntrimmedNets for Weakly Supervised Action Recognition and Detection. In *CVPR*. 6402–6411.

[25] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming Ming Cheng, Zhao Yao, and Shuicheng Yan. 2017. Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach. In *CVPR*.

[26] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. 2017. A Pursuit of Temporal Accuracy in General Activity Detection. *CoRR* abs/1703.02716 (2017).

[27] Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *ICCV*. 5783–5792.

[28] Yunlu Xu, Chengwei Zhang, Zhanzhan Cheng, Jianwen Xie, Yi Niu, Shiliang Pu, and Fei Wu. 2019. Segregated Temporal Assembly Recurrent Networks for Weakly Supervised Multiple Action Detection. *AAAI*.

[29] Ke Yang, Peng Qiao, Dongsheng Li, Shaohe Lv, and Yong Dou. 2018. Exploring Temporal Preservation Networks for Precise Temporal Action Localization. In *AAAI*. 7477–7484.

[30] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal Action Detection with Structured Segment Networks. In *ICCV*. 2933–2942.

[31] Jia Xing Zhong, Nannan Li, Weijie Kong, Zhang Tao, and Li Ge. 2018. Step-by-step Erasion, One-by-one Collection: A Weakly Supervised Temporal Action Detector. In *ACM Multimedia Conference*. 35–44.

[32] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*. 2921–2929.