Segregated Temporal Assembly Recurrent Networks for Weakly Supervised Multiple Action Detection

Yunlu Xu,[1*] Chengwei Zhang,[2*] Zhanzhan Cheng,[13†] Jianwen Xie,[1] Yi Niu,[1] Shiliang Pu,[1] Fei Wu[3]
[1]Hikvision Research Institute, China; [2]Shanghai Jiaotong University, China; [3]Zhejiang University, China
{xuyunlu,chengzhanzhan,jianwen.xie,niuyi,pushiliang}@hikvision.com; cwzhang@sjtu.edu.cn; wufei@cs.zju.edu.cn

Abstract

This paper proposes a segregated temporal assembly recurrent (STAR) network for weakly-supervised multiple action detection. The model learns from untrimmed videos with only supervision of video-level labels and makes prediction of intervals of multiple actions. Specifically, we first assemble video clips according to class labels by an attention mechanism that learns class-variable attention weights and thus helps the noise relieving from background or other actions. Secondly, we build temporal relationship between actions by feeding the assembled features into an enhanced recurrent neural network. Finally, we transform the output of recurrent neural network into the corresponding action distribution. In order to generate more precise temporal proposals, we design a score term called segregated temporal gradient-weighted class activation mapping (ST-GradCAM) fused with attention weights. Experiments on THUMOS'14 and ActivityNet1.3 datasets show that our approach outperforms the state-of-the-art weakly-supervised method, and performs at par with the fully-supervised counterparts.

1 Introduction

Multiple action detection, which aims at localizing temporal intervals of actions and simultaneously identifying their categories in videos, is a fundamental problem in video understanding. Many existing works (Shou, Wang, and Chang 2016; Zhao et al. 2017; Shou et al. 2017; Xu, Das, and Saenko 2017; Yang et al. 2018; Chao et al. 2018; Lin et al. 2018; Alwassel, Caba Heilbron, and Ghanem 2018) make efforts to address this problem in a supervised manner, where the algorithms rely on fully labeled data (i.e., videos with precise annotations of the starting and ending frames of actions). However, such supervised methods are prohibitively impractical in real applications, since frame-level annotations are substantially time-consuming and expensive. Therefore, learning to detect temporal action from untrimmed videos remains a crucial and challenging problem in video understanding.

A few explorations based solely on video-level annotations have exemplified the weakly supervised temporal action detection. UntrimmedNet (Wang et al. 2017) proposes to learn a selection module for detecting important segments, and later STPN (Nguyen et al. 2018) conquers the single-label limitation by introducing temporal class activation maps (T-CAM) trained with cross-entropy loss. To address the issue that performing localization via thresholds may not be robust to noises in class activation maps, AutoLoc (Shou et al. 2018) directly predicts temporal boundary and proposes a Outer-Inner-Contrastive loss to provide the desired segment-level supervision. W-TALC (Paul et al. 2018) introduces the Co-Activity Similarity Loss and jointly optimizes it with the cross-entropy loss for the weakly-supervised temporal action detection. However, the interference and relationship among actions in a video hitherto have not been concerned.

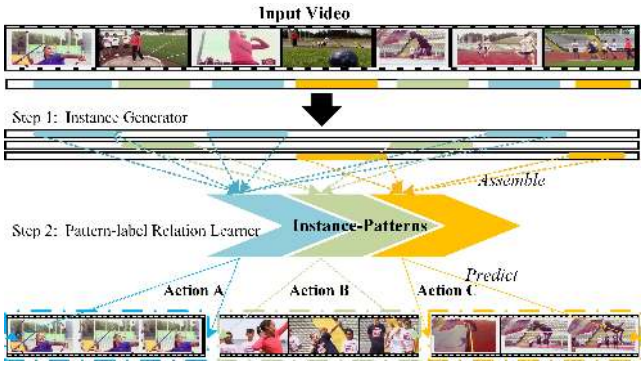

Figure 1: An illustration of the multiple action detection task from the perspective of MIML.

In reality, a video in general describes multiple actions occurring in a complex background. Intuitively, a desired video descriptor should have two characteristics: (1) refraining from the interference of other unrelated actions or background, and (2) enhancing the correlation among actions.

In this paper, we focus on the task of weakly supervised multiple action detection with only video-level labels. As illustrated in Figure 1, the task of multiple action detection in a weakly-annotated video can be regarded as a multi-instance multi-label (MIML) (Zhou et al. 2008) problem where an example (i.e., a video) is described by multiple instances (i.e., actions) and associated with multiple class

*Authors contribute equally. Zhang did this work during an internship in Hikvision Research Institute.

†Corresponding author.

labels (i.e., action categories). Correspondingly, we address the weakly supervised action detection task in the following two steps. Step 1: Action assembling for multi-instance pattern generation. In order to eliminate interference from unrelated actions or complex background, we generate instance-patterns for each type of action via an action selector. As shown in Figure 1, three intervals of *Action-A* (denoted in blue) are selected and integrated as an assembled feature representation *Instance-pattern-A*. By this way, each instance-pattern is mapped from the corresponding type of action in the input video. Step 2: Relation learning for label generation. With assembled patterns, the corresponding instances can be directly predicted, but some correlation (*e.g., CricketShot* always co-occurs with *CricketBowling*) generally exists in the case of multiple actions in a video. Therefore, we learn the implicit relationship between different instances by adopting a recurrent neural network (Hochreiter and Schmidhuber 1997). Furthermore, the corresponding instance proposals can be activated from the learned class-variable weights.

More specifically, we propose a weakly-supervised framework for multiple temporal action detection called **S**egregated **T**emporal **A**ssembly **R**ecurrent (*abbr*. STAR) network. Firstly, we construct a well-designed attention module to learn the action *assembly weights* for integrating the encoded segmented features into corresponding instance-patterns. Then we learn the relationship between instances by adopting an enhanced recurrent neural network (RNN) for action label generating. We also involves a *repetition align mechanism* in RNN for adaptively adjusting the attention weights to generate finer action proposals. Finally, we design an operation term called segregated temporal Gradient-CAM (ST-GradCAM), which is an extension of Gradient-CAM (Selvaraju et al. 2017), to indicate feature significance for a specific action category. We fuse the response of ST-GradCAM with the learned assembly weights for the purpose of action localization.

The contributions of our paper are four-fold: (1) We reformulate the multiple action detection from a MIML perspective, i.e., extracting instance-patterns and generating action labels, which eliminates interference among unrelated action features and captures temporal dependency between multiple concurrent actions. (2) An end-to-end framework called STAR, which includes a well-designed attention module and an enhanced RNN, is developed to be trained in a weakly supervised manner from videos with only video-level labels. (3) We design an ST-GradCAM operation fused with class-variable assembly weights for action temporal localization. (4) Experiments demonstrate that our weakly supervised framework achieves impressive performance on the challenging THUMOS'14 (Jiang et al. 2014) and ActivityNet1.3 (Heilbron et al. 2015) datasets for action detection, comparable with those of supervised learning methods.

2 Related Work

Action Recognition. The task of action recognition seeks to identify a single or multiple action labels for each video and is often treated as a classification problem. Before the era of deep learning, hand-crafted features, such as the improved dense trajectories (Wang and Schmid 2013), obtained outstanding performance on many benchmark datasets. Recently, there have been vast works on action recognition using convolutional neural networks (CNN). For example, a 2D CNN for large-scale video classification was first investigated in (Karpathy et al. 2014), but has not achieved comparable performance with hand-crafted features. Two-stream (Simonyan and Zisserman 2014) and C3D (Tran et al. 2015; 2018; Carreira and Zisserman 2017) networks are recent mainstreams to learn discriminative features for action recognition. The inception 3D (I3D) (Carreira and Zisserman 2017) is a two-stream network based on a 3D version of Inception network (Ioffe and Szegedy 2015), which is commonly used as a feature encoder for action localization (Nguyen et al. 2018) and dense-labeling videos (Piergiovanni and Ryoo 2018).

Fully Supervised Action Detection. Different from action recognition, action detection aims to identify the temporal intervals containing target actions. Most existing works focus on fully-supervised approaches for that. To capture robust video feature representation, S-CNN (Shou, Wang, and Chang 2016) uses a multi-stage CNN for temporal action localization. SSN (Zhao et al. 2017) introduces structured temporal pyramids with decoupled classifiers for classifying actions and determining completeness. For precise boundaries, the Convolutional-De-Convolutional (CDC) network (Shou et al. 2017) and the Temporal Preservation Convolutional (TPC) network (Yang et al. 2018) are proposed for frame-level action predictions. Boundary Sensitive network (BSN) (Lin et al. 2018) is recently proposed to locate temporal boundaries which are further integrated into action proposals. Furthermore, some region-based methods, e.g., R-C3D (Xu, Das, and Saenko 2017) and TAL-Net (Chao et al. 2018), propose to generalize the methods for 2D spatial detection to 1D temporal localization.

Weakly Supervised Action Detection. Action detection in a weakly supervised fashion has been studied by only a few works. UntrimmedNet (Wang et al. 2017) is an end-to-end model for learning single-label action classification and action detections. Hide-and-seek (Singh and Yong 2017) tries to force the model to see different parts of the image and focus on multiple relevant parts of the object beyond just the most discriminative one by randomly masking different regions of training images in each training epoch. Such a method works well for spatial object detection but is unsatisfied for the temporal action detection. STPN(Nguyen et al. 2018) adopts an attention module to identify a sparse subset of key segments associated with target actions in a video, and fuse the key segments via adaptive temporal pooling. The latest work (Shou et al. 2018) and (Paul et al. 2018) boost the STPN by introducing novel objective functions to separately tune coarse action boundaries and unearth co-activity relationship between videos.

Multi-instance Multi-label framework. Single-instance single-label (Krizhevsky, Sutskever, and Hinton 2012) framework has led to remarkable performance. While in the real-world settings, an example is usually composed of multiple instances, such as sentences in a text, image frames in a video and objects in an image. Multi-instance multi-label

(MIML) (Zhou et al. 2008) framework, where an example is described by multiple instances and associated with multiple class labels, has been applied to different tasks, such as multi-label image classification (Wang et al. 2016), image retrieval (Zhang et al. 2018), object detection and semantic segmentation (Wei et al. 2017; Ge, Yang, and Yu 2018), and sound separation (Gao, Feris, and Grauman 2018). Our paper generalizes this framework to the action detection task where a video contains multiple actions and is associated with multiple video-level classes.

3 Segregated Temporal Assembly Network

The framework of STAR, which is shown in Figure 2, consists of three components: (1) a pre-trained feature extractor for encoding a video into a sequence of segmental feature vectors, (2) an end-to-end trainable architecture that we call segregated temporal assembly network, including an action assembler and a label generator, and (3) a well-designed action localizor for action location. In the following, we first present the architecture of the proposed STAR in *Section 3.1*; Then a well-designed action proposal mechanism is described in *Section 3.2*; Finally, the training strategy of the whole network is given in *Section 3.3*.

3.1 Action Assembly and Label Generation

Stem Architecture Given N segments of K-dimensional feature vectors $S=\{s_1, s_2, \ldots, s_N\} \in \mathbb{R}^{K\times N}$, which are extracted from a video $\mathcal{V}$ by a pre-trained feature extractor, we first assemble actions from S into instance-patterns $X=\{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{K\times T}$, where T is the number of assembly actions in $\mathcal{V}$. Then we use an RNN to build relation between the assembled actions in X, and further generate the action labels y_i from a label set $Y = \{y_1, y_2, \ldots, y_T\}$ one-by-one. Concretely, we first assemble actions into a specific instance-pattern at time t by

$$x_t = \sum_{i=1}^{N} \alpha_{t,i} s_i, \tag{1}$$

where α is the learnt attending assembly weights over S. Generally, α is calculated by simultaneously referring the last hidden states of RNN and glimpsing the whole input S (Chorowski et al. 2015). Correspondingly, we first evaluate the energy state e over S by

$$e_{t,i} = v_\alpha \varphi(W_\alpha h_{t-1} + U_\alpha s_i), \tag{2}$$

where h_{t-1} is the hidden state of RNN at time t-1; Then the energy state is further normalized by

$$\alpha_{t,i} = \sigma(e_{t,i}), \tag{3}$$

where φ and σ are the activation function *tanh* and *sigmoid* respectively, and W_α, U_α and v_α are learnable parameters. Note that, the conventional attention mechanism (Chorowski et al. 2015) uses *softmax* function to normalize the energy distribution, which results in failures to capture those long or high-frequency actions. Instead of *softmax*, we adopt the bounded logistic sigmoid activation function σ on the energy distribution to deal with this issue. Figure 3 illustrates a comparison of effects of different activation functions.

After assembling actions from S to X, we use the RNN to build relation between instance-patterns in X by

$$h_t = RNN(h_{t-1}, y_{t-1}, x_t), \tag{4}$$

where RNN is specified as the popular relation learning model long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997). Then we further output the action probability distribution by

$$y_t = softmax(W_o h_t), \tag{5}$$

where W_o is the learnt parameters.

Though the above process, denoted as the naive stem network, can generate action labels to a certain extent, it is still unsatisfactory to handle complicated video action tasks due to the following three key factors:

1. *Attending repetition*: Repetition of attending regions is a common problem for sequence-to-sequence models (Tu et al. 2016) and is especially pronounced when generating multiple instance-patterns (see *Raw Attention* in Figure 4), which goes against the purity of single pattern.
2. *Co-occurrence*: Unlike usual sequential applications (e.g., text reading, speech recognition, translation etc.), co-occurrence is universal in videos (*e.g., CricketBowling* and *CricketShot* always appear successively and have overlapping), which increases the challenge of video tasks.
3. *Trivial action missing*: Some repetitive but inapparent action features are usually shielded by the corresponding prominent representative patterns due to the lack of frame-level annotations, leading to failures of trivial action detection.

Permissive Coverage Constraint For *Attending repetition*, coverage mechanism (Tu et al. 2016) has been introduced to minimize the overlapping of attention weights across time steps, assuming that once given high score-weight in one step, the input vector must not be focused in the future steps. However, shown as *Original Coverage* in Figure 4, conventional coverage mechanism strictly forbids focus on the same place, which is not suitable for video task because of action *co-occurrences* phenomenon. Instead, we design an action-friendly *permissive coverage* constraints on the weights, in which values of a certain step not only are constraint to the previous weights, but also refer to the last hidden state, which is shown as *Permissive Coverage* in Figure 4. Specifically, we rewrite the coverage score at t-th step for segment s_i as

$$\begin{aligned} COV_{t,i} &= f(h_{t-1}, \alpha_{t-1,i}) \\ &= \sigma[Z_i(i - \sum_{k=1}^{N} \alpha_{t-1,k} k) + W_\alpha h_{t-1}], \end{aligned} \tag{6}$$

where Z_i is the corresponding learnable parameter, and f is a nonlinear activation function composed of a MLP structure and the *sigmoid* activation. The $COV_{t,i}$ simultaneously restricts the current weight α_t attending relevant to α_{t-1} and refers to the last recurrent hidden state h_{t-1}.

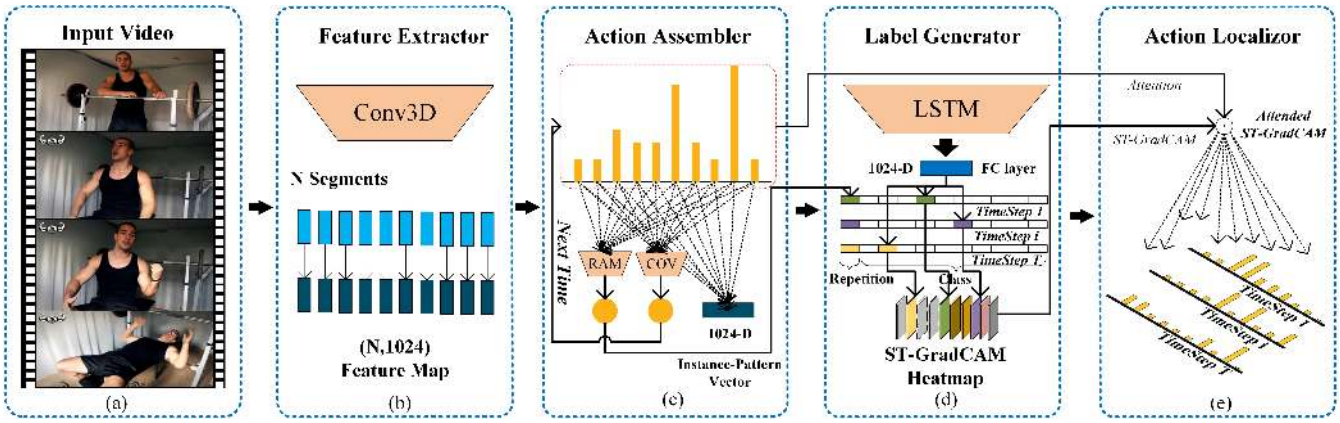

Figure 2: The workflow of the STAR framework. (a) The input video; (b) A pre-trained video encoder for segmental feature extraction; (c) An action assembler for generating instance-patterns, in which stage attention weights are trained with well-designed sub-modules (e.g., RAM); (d) A LSTM-based network for action label generation ; (e) A localizer for locating actions from input video, only used for inference without any training.

(a) softmax (b) sigmoid

Figure 3: Effects of different normalization functions on energy distribution for attending actions. Horizontal and vertical axes separately represent time and energy score. The ground truth of the action regions is represented by a bar, with yellow color indicating action occurrence.

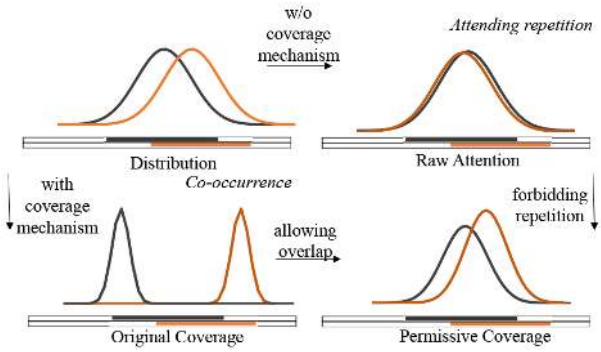

Figure 4: Attending weights distribution in different *coverage* mechanisms. Horizontal and vertical axes separately stand for time and energy score. The *black* and *orange* bars refer to the ground truth of two different actions. Curves in *black* and *orange* are corresponding to the energy distributions in two steps.

Then we put the coverage term into the attending mechanism by rewriting Equation 2 as

$$\hat{e}_{t,i} = v_\alpha[COV_{t,i}\varphi(W_\alpha h_{t-1} + U_\alpha s_i)], \tag{7}$$

where $COV_{t,i}$ can be considered as a *soft gating value* within the range of [0,1].

Repetition Alignment Mechanism For *trivial action missing*, we propose a repetition alignment module (RAM) to calculate the frequency of single-instance occurring in each video, which heuristically tunes the action proposal generation. Namely, RAM not only manifests trivial actions occurring, but also restrains the unrelated time segments. The RAM at t-th step is calculated by

$$RAM_t = W_r \sum_{i=1}^{N} \sigma(\hat{e}_{t,i}), \tag{8}$$

where W_r is the learned parameters. RAM_t is supervised with the number of corresponding action frequency. Then we further involve this term into the RNN structures for enhancing the pattern-label relation learning. Thus Equation 4 is extended as

$$h_t = RNN(h_{t-1}, y_{t-1}, x_t, RAM_t). \tag{9}$$

Note that, counting the occurrence of each action category need neither the frame-level information nor precise time locations. RAM is an effective and flexible assistant sub-module in the whole action assembly generation.

3.2 Action Proposal Generation

The Class Activation Mapping (CAM) (Zhou et al. 2016) is useful for action localization, and also has been applied in the previous work (Nguyen et al. 2018). However, CAM is only designed for linear architectures, and not suitable for nonlinear architectures, such as RNN. While Gradient-CAM (Grad-CAM) (Selvaraju et al. 2017) is applicable to any differentiable architecture even with activations. In this work, we adopt a more general Grad-CAM to calculate class response for our task, termed as Segregated Temporal Gradient-weighted Class Activation Map (ST-GradCAM).

At the t-th step, the prediction output d_t^c (output distribu-

tion for a class *c* before the *softmax*) is represented by

$$d_t^c = \sum_{k=1}^{K} w_{t,k}^c x_t^k, \tag{10}$$

where $w_{t,k}^c$ is the importance of the k-th feature value x_t^k for a target class *c*, which is represented by the following gradient score

$$w_{t,k}^c = \frac{\partial d_t^c}{\partial x_t^k} = \frac{\partial d_t^c}{\partial h_t^c} \cdot \frac{\partial h_t^c}{\partial x_t^k} = W_o \cdot \frac{\partial RNN(x_t)}{\partial x_t^k}, \tag{11}$$

where h_t^c is the importance of h_t for the target class *c*. Since the attention weights possess rich information regarding action intervals (Wang et al. 2017; Nguyen et al. 2018), we formulate Equation 10 as

$$d_t^c = \sum_{k=1}^{K} w_{t,k}^c (\sum_{i=1}^{N} \alpha_{t,i} s_i^k) = \sum_{i=1}^{N} \alpha_{t,i} \sum_{k=1}^{K} w_{t,k}^c s_i^k, \tag{12}$$

where s_i^k is the k^{th} feature value s_i^k in s_i.

Since d_t^c indicates the importance of representations to each class at recurrent step *t*, a class-aware activation map can be derived from above. We define ST-GradCAM as

$$\xi_{t,i}^c = \sum_{k=1}^{K} w_{t,k}^c s_i^k, \tag{13}$$

where i indexes the segment in S. ST-GradCAM captures the important local information of feature map *k* for a target class *c* at recurrent step *t*.

To generate temporal action proposals, we train a two-stream network and derive the attended ST-GradCAM at t-th step using $\alpha_{t,i} \cdot \sigma(\xi_{t,i}^c)$. For each class *c* at the recurrent step *t*, each proposal $[N_{start}, N_{end}]$ is assigned a score by:

$$\sum_{i=N_{start}}^{N_{end}} \frac{[\lambda \cdot \alpha_{t,i,RGB}^c + (1-\lambda) \cdot \alpha_{t,i,flow}^c]}{N_{end} - N_{start} + 1} \cdot \sigma(\xi_{t,i}^c), \tag{14}$$

where we fuse the attention values of RGB and optical flow streams by the modality ratio λ ($\lambda = 0.5$ by default) first, and then generate proposals based on RGB and flow separately. For final detection, we perform non-maximum suppression (NMS) among temporal proposals of each class by removing highly overlapped ones.

3.3 Network Training

The training objective of the STAR network is to solve a multi-task optimization problem. The overall loss function consists of four terms: the classification loss, coverage loss, repetition alignment loss and sparsity loss,

$$L = L_{class} + \beta L_{sparsity} + \gamma L_{cov} + \delta L_{ram}, \tag{15}$$

where γ, δ and β are the hyper-parameters.

Given encoded segment inputs S, *classification loss* is defined as the softmax loss over multiple categories by

$$L_{class} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{T} \log P_i(\hat{y}_t | S, \theta), \tag{16}$$

where M, $\hat{y}_t$ and θ represent the number of training videos, the ground truth of the t-th action category, and all the trainable parameters respectively. P_i is the multinomial logistic regression (a probability density over all action categories).

The *coverage loss* is to overcome the common *laziness* of learning problems and thus to put emphasis on different action segments, which is computed by

$$L_{cov} = max(0, \sum_{i=1}^{N} (\sum_{k=1}^{i} \alpha_{t,k} - \sum_{k=1}^{i} \alpha_{t-1,k})). \tag{17}$$

The *RAM loss* is designed to relieve the *trivial action missing* problem by checking the repetition number, which adopts the L2 loss and is defined as

$$L_{ram} = \frac{1}{2T} \sum_{t=1}^{T} ||c_t - RAM_t||_2^2, \tag{18}$$

where c_t is the ground-truth of the t-th action frequency.

The *sparsity loss* $L_{sparisity}$ is the L1 regularization on the attention weights, i.e., $||\alpha||_1$.

4 Experiments

We evaluate our proposed framework (STAR) with mean average precision (mAP) score on two benchmarks for temporal action detection, i.e., THUMOS'14 (Jiang et al. 2014) and ActivityNet1.3 (Heilbron et al. 2015). Following the routine evaluation protocol in (Nguyen et al. 2018), our method outperforms existing weakly-supervised methods.

4.1 Implementation Details

Datasets. *THUMOS'14*, extracted from over 20 hours of sport videos, consists of 20 action classes. It contains 200 videos from validation set for training, and 212 videos for testing. This dataset is challenging for temporal detection because (1) averagely, each video contains more than 15 occurrences of all actions, (2) the length of an action varies significantly (e.g., from less than one second to over 26 minutes), and (3) averagely, each video is with about 71% background. It is a good benchmark for multiple action detection.

ActivityNet1.3 contains 200 activity categories, in which 10,024 videos are used for training, 4,926 for validation, and 5,044 for testing. *ActivityNet1.3* only contains 1.5 occurrences per video on average and most videos simply contain single action category with averagely 36% background.

Training Details. Our model is implemented on Caffe. For a direct and fair comparison, we follow the video preprocessing procedure of STPN (Nguyen et al. 2018) by pretraining the two-stream I3D network (Carreira and Zisserman 2017) on Kinetics dataset (Kay et al. 2017). Then we uniformly sample 400 segments from each video for feature extraction. The whole network is trained by using Adam optimizer with learning rate 10^{-4} and dropout ratio 0.8 on both streams. Besides, β, γ and δ in Equation 15 are empirically set to be 10^{-4}, 10^{-4} and 10^{-6} respectively.

Testing Details. We retrieve one-dimensional temporal proposals from the predicted label distribution d based on the outputs of the two-streams network. As the two streams have similar classification performance, we set a modality ratio of 1:1 (RGB:flow) as classification confidence scores and make the prediction jointly. Then we exploit Equation 14 to output the action proposals.

4.2 Ablation Study

To analyze the contributions of several different components of STAR, we conduct the ablation study on the THUMOS'14 dataset. Performance is evaluated with average mAP (%) by calculating the multiple overlap IoU with thresholds varying from 0.1 to 0.5.

Effects of architecture modules We investigate modules including *coverage* constraints (COV), *sparsity* (SPA), and *RAM* with *stem* network of STAR. Table 1 shows the effects of each module and their combinations.

- *Stem network*. It serves as our baseline model.
- *Stem with one module*. In overall, each single module can improve the stem structure by 3%-5%. Interestingly, both (SPA) and (RAM) are constraints on the frequency and extent of attended weights, and though by different means, they obtain similar performance alone, better than (COV). We can infer that direct penalty of occurrence is more effective than the handling of action co-occurrence.
- *Stem with two modules*. RAM with either *sparsity* or *coverage* can achieve better performance than other components, which indicates that additional repetition information is very useful. (SPA, COV) is unsatisfying compared to other module combinations, even worse than (SPA) solely. We analyze the reason that both of them restrains the extents of attended weights, but (RAM) aligns the extents by zooming in and out. So combinations with (RAM) achieve steady improvements, while the combination without (RAM) tends to suppress excessively.
- *Stem with all modules*. STAR with all modules achieves the best performance which improve the *stem* by 8%.

Table 1: Performance evaluation with different modules of STAR on THUMOS'14.

Stem	√	√	√	√	√	√	√	√
Sparsity		√			√	√		√
Coverage			√		√		√	√
RAM				√		√	√	√
Ave-mAP(%)	39.0	43.8	42.4	43.8	43.3	44.0	44.7	**47.0**

Effects of detection operations As introduced in Equation 12, ST-GradCAM and the learned *assembly weights* are respectively responsible for indication of class-specific contribution and input segmented duration. Since the learned attention weights are sensitive to action classes, we can alternatively use only attention weights without ST-GradCAM to propose locations of each class, termed by *Attention*. Similarly, the location of each class also can be proposed based on only ST-GradCAM without attention weights,(i.e., *ST-GradCAM*). We also integrate both learned attention weights and ST-GradCAM as a fused action detector, denoted by *Attended ST-GradCAM* (used by default in our work).

Table 2: Effects of different modules on THUMOS'14.

Weakly-supervised Model	Ave-mAP(%)
Wang et al. (2017)	29.0
Singh et al. (2017)	20.6
Nguyen et al. (2018)	35.0
Paul et at.(2018)	39.7
ST-GradCAM	24.4
Attention	39.6
Attended ST-GradCAM	**47.0**

Table 2 gives the results. We find that $Attention$ already has achieved comparable performance to previous methods, implying that the learned attention weights themselves contain rich location information and play important roles in the entire detection process. As expected, the *Attended ST-GradCAM* significantly outperforms each single term (*i.e*, *Attention* and *ST-GradCAM*), which demonstrates the effectiveness of the STAR scoring mechanism (in Equation 10).

4.3 Qualitative Evaluation

STAR can iteratively segregate different actions from the origin input video segments, then assemble the corresponding actions into a target action-patterns. For further demonstrating the performance of STAR, we qualitatively analyze the effects of STAR from different aspects as follows:

- *Effect of Attention*: The learnt assembly weights are used to assemble actions into the corresponding instance-patterns so that the weights are capable of indicating action locations (*e.g.,* see intervals $[16.0s, 17.6s]$ and $[17.6s, 18.1s]$ in Figure 5), in consonance with the performance in subsection *ablation study*. However, it still suffers from ambiguous boundaries of actions (*e.g.* see time $48.4s$ or $85.6s$ in Figure 5).
- *Effect of Attended ST-GradCAM*: Considering the response mechanism of ST-GradCAM, the *Attended ST-GradCAM* can achieve more precise action location than *Attention* (*e.g.* at time $85.6s$ or $103.7s$ in Figure 5).
- *False Positive Analysis*: We also analyze the false positives, and find that those falsely detected image frames usually bear high similarities to the annotations, (*e.g.* actions before time $84.9s$ or after time $86.4s$ in Figure 5) and are even ambiguous to human beings.
- *Evaluation of Extreme Scenarios*: In general, action may occur sparsely or densely in videos, and the degree to which a video is filled with actions can be measured by the action occurring *density*, which is defined as the overlap of all action intervals over the given whole video. For example, $denisty = 0$ means that there is no actions occurring in the video, while $denisty = 1$ means actions occur continuously in the whole video. Methods for action detection easily fail in videos with excessively either

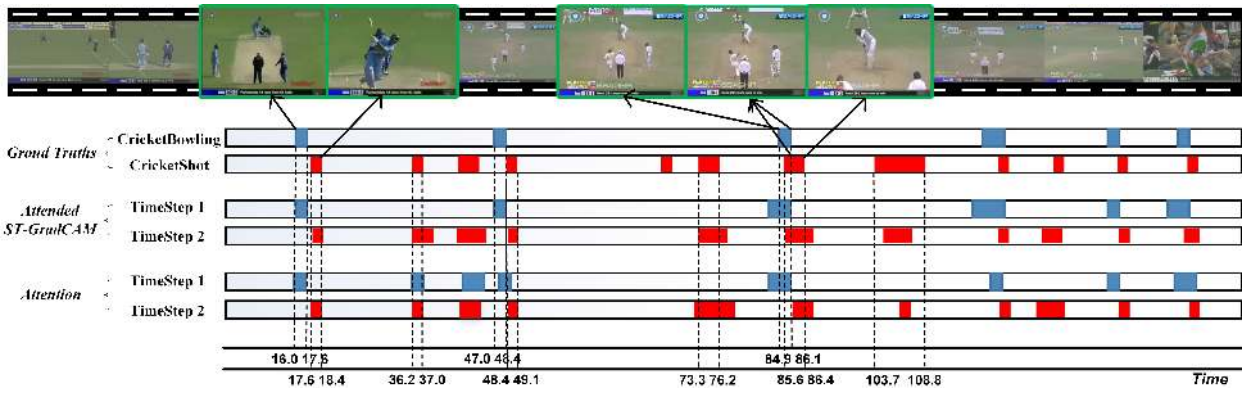

Figure 5: An example for action localization on THUMOS'14, which contains two actions (*CricketBowling* denoted and *CricketShot*). The video is segregated into two assemblies (TimeStep1 and TimeStep2) step-by-step.

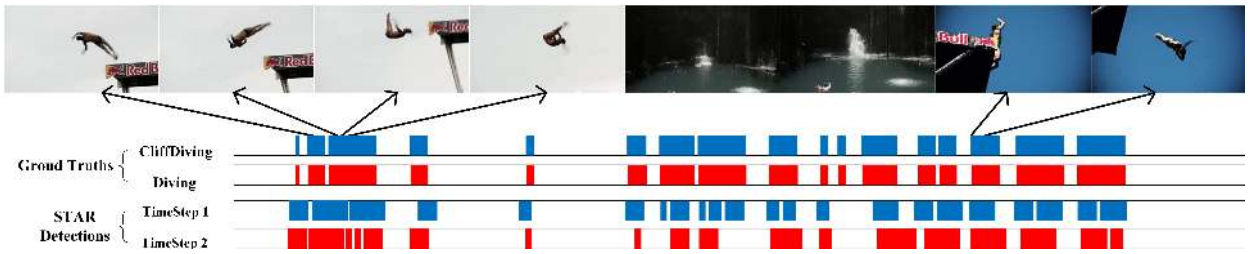

Figure 6: Results on videos persistently occurring multi-actions.

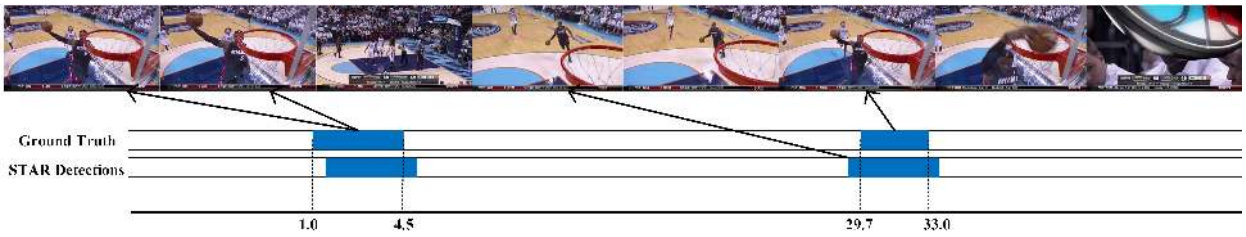

Figure 7: Results on videos with sparse single-actions.

sparse or dense action occurrence. Figure 6 and 7 display the action detection results on *dense* and *sparse* situations respectively, which further demonstrate the robustness of our method. We also report the performances in terms of Ave-mAP with different densities of occurring actions in videos, which is shown in Figure 8. The results show that STAR maintains high performances under different levels of action occurring density.

4.4 State-of-the-Art Comparisons

We compare STAR with state-of-the-art weakly-supervised and fully-supervised methods on THUMOS'14 and ActivityNet1.3 datasets. Note that, THUMOS'14 is a better benchmark for evaluating our method, as addressed in *Datasets* section. Table 3 and 4 summarize the results.

Comparison with weakly supervised methods. It is shown that STAR outperforms all other weakly supervised methods by a remarkably large margin, improving the reported highest average mAP about 7% by 47.0% on THUMOS'14 and 2% by 18.1% on ActivityNet1.3. Note that, on THUMOS'14, our model achieves more than 10% higher than existing methods on both IoU 0.1 and 0.2. This verifies the superiority of our framework.

Comparison with fully supervised methods. In Table 3, our model also has comparable results with those fully supervised methods. There is a great gap between the existing fully and weakly supervised methods because of the usages of detailed boundary annotations in fully supervised methods. However, our model still outperforms all fully supervision results at both IoU 0.1 and 0.2 on THUMOS'14 dataset.

As in Table 4, results with asterisk (*) are collected from ActivityNet Challenge submissions (only for general references here), which can not be impartially compared directly with our STAR. We see that our model even overtakes the recent strong-supervised method (Xu, Das, and Saenko 2017) and partially surpasses method (Chao et al. 2018), but falls behind work BSN (Lin et al. 2018). Note that, BSN takes full use of boundary annotations via a sophisti-

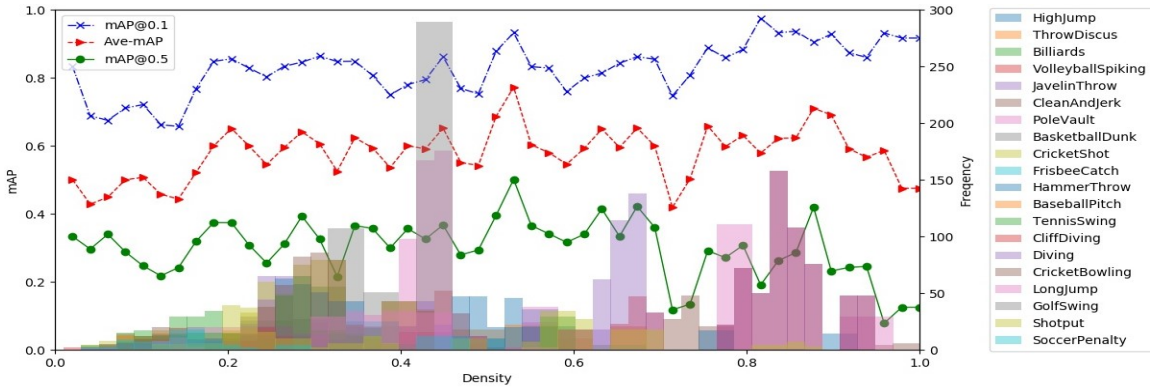

Figure 8: The performance (mAP) with varying action *density* values on THUMOS'14. The horizontal, left-vertical and right-vertical axes separately represent the density of action occurring, the mAP and the frequency of action occurring. Three curves describe the mAP performances with IoU 0.1 and IoU 0.5, and the average IoU over 5 thresholds within $[0.1, 0.5]$. The colored (denoted as different type of actions) histogram refers to the frequency of corresponding actions occurring at the specific density.

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

Supervision	Method	AP@IoU 0.1	0.2	0.3	0.4	0.5
Fully Supervised	Richard (2016)	39.7	35.7	30.0	23.2	15.2
	Shou (2016)	47.7	43.5	36.3	28.7	19.0
	Yeung (2016)	48.9	44.0	36.0	26.4	17.1
	Yuan (2016)	51.4	42.6	33.6	26.1	18.8
	Shou (2017)	–	–	40.1	29.4	23.3
	Yuan (2017)	51.0	45.2	36.5	27.8	17.8
	Gao (2017)	54.0	50.9	44.1	34.9	25.6
	Xu (2017)	54.5	51.5	44.8	35.6	28.9
	Zhao (2017)	**66.0**	**59.4**	51.9	41.0	29.8
	Lin (2017)	50.1	47.8	43.0	35.0	24.6
	Yang (2018)	–	–	44.1	37.1	28.2
	Chao (2018)	59.8	57.1	53.2	**48.5**	**42.8**
	Alwasssel (2018)	49.6	44.3	38.1	28.4	19.8
	Lin (2018)	–	–	**53.5**	45.0	36.9
Weakly Supervised	Wang (2017)	44.4	37.7	28.2	21.1	13.7
	Singh (2017)	36.4	27.8	19.5	12.7	6.8
	Nguyen (2018)	52.0	44.7	35.5	25.8	16.9
	Shou (2018)	–	–	35.8	29.0	21.2
	Paul (2018)	55.2	49.6	40.1	31.1	22.8
	Ours	**68.8**	**60.0**	**48.7**	**34.7**	**23.0**

cated multi-stage training strategy. In conclusion, STAR surpasses all the reported weakly-supervised methods on both two benchmarks. Although fully-supervised approaches still have good results at large IoU thresholds, STAR significantly narrows down the gap between the fully and weakly supervised methods.

5 Conclusion

We propose an end-to-end weakly supervised framework STAR for action detection in MIML perspective. The model first assembles actions into corresponding instance-patterns with a well-designed attention mechanism, and then learns the temporal relationship between multiple instance-patterns by using RNN. Finally, with the predicted action labels and the learned attention weights, we use a well designed ST-GradCAM for localizing each action. Experiments show that our approach outperforms all the reported results by weakly supervised approaches by a large margin, and also achieves comparable performance with those fully supervised methods on both THUMOS'14 and ActivityNet1.3 datasets.

Table 4: A comparison on ActivityNet v1.3 validation set. The sign (*) indicates results from ActivityNet Challenge.

Supervision	Method	AP@IoU 0.5	0.75	0.95
Fully Supervised	Singh (2016)*	34.5	–	–
	Shou (2017)*	45.3	26.0	0.2
	Dai (2017)*	36.4	21.2	3.9
	Xiong (2017)*	39.1	23.5	5.5
	Lin (2017)*	49.0	32.9	7.9
	Xu (2017)	26.8	–	–
	Chao (2018)	38.2	18.3	1.3
	Lin (2018)	**52.5**	**33.5**	**8.9**
Weakly Supervised	Nguyen (2018)	29.3	16.9	2.6
	Ours	**31.1**	**18.8**	**4.7**

References

Alwassel, H.; Caba Heilbron, F.; and Ghanem, B. 2018. Action Search: Spotting Actions in Videos and Its Application to Temporal Action Localization. In *ECCV*, 251–266.

Carreira, J., and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 4724–4733.

Chao, Y. W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*, 2933–2942.

Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *NIPS*, 577–585.

Dai, X.; Singh, B.; Zhang, G.; Davis, L. S.; and Chen, Y. Q. 2017. Temporal Context Network for Activity Localization in Videos. In *ICCV*, 5727–5736.

Gao, R.; Feris, R.; and Grauman, K. 2018. Learning to Separate Object Sounds by Watching Unlabeled Video. In *ECCV*, 35–53.

Gao, J.; Yang, Z.; and Nevatia, R. 2017. Cascaded Boundary Regression for Temporal Action Detection. *CoRR* abs/1705.01180.

Ge, W.; Yang, S.; and Yu, Y. 2018. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning. In *CVPR*, 1277–1286.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Ioffe, S., and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 448–456.

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

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. F. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *CVPR*, 1725–1732.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *ECCV*, 3–19.

Lin, T.; Zhao, X.; and Shou, Z. 2017. Single Shot Temporal Action Detection. In *ACM MM*, 988–996.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly Supervised Action Localization by Sparse Temporal Pooling Network. In *CVPR*, 6752–6761.

Paul, S.; Roy, S.; and Roy Chowdhury, A. K. 2018. W-TALC: Weakly-supervised Temporal Activity Localization and Classification. In *ECCV*, 563–579.

Piergiovanni, A. J., and Ryoo, M. S. 2018. Learning Latent Super-Events to Detect Multiple Activities in Videos. In *CVPR*, 5304–5313.

Richard, A., and Gall, J. 2016. Temporal Action Detection Using a Statistical Language Model. In *CVPR*, 3131–3140.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*, 618–626.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*, 5734–5743.

Shou, Z.; Gao, H.; Zhang, L.; Miyazawa, K.; and Chang, S.-F. 2018. AutoLoc: Weakly-supervised Temporal Action Localization in Untrimmed Videos. In *ECCV*, 154–171.

Shou, Z.; Wang, D.; and Chang, S. F. 2016. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *CVPR*, 1049–1058.

Simonyan, K., and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*, 568–576.

Singh, G., and Cuzzolin, F. 2016. Untrimmed Video Classification for Activity Detection: submission to ActivityNet Challenge. *CoRR* abs/1607.01979.

Singh, K. K., and Yong, J. L. 2017. Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization. In *ICCV*, 3544–3553.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 6450–6459.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016. Modeling Coverage for Neural Machine Translation. In *ACL*, 76–85.

Wang, H., and Schmid, C. 2013. Action recognition with improved trajectories. In *ICCV*, 3551–3558.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *CVPR*, 2285–2294.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. UntrimmedNets for Weakly Supervised Action Recognition and Detection. In *CVPR*, 6402–6411.

Wei, Y.; Liang, X.; Chen, Y.; Shen, X.; Cheng, M.; Feng, J.; Zhao, Y.; and Yan, S. 2017. STC: A Simple to Complex Framework for Weakly-Supervised Semantic Segmentation. *IEEE TPAMI* 39(11):2314–2320.

Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; and Tang, X. 2017. A Pursuit of Temporal Accuracy in General Activity Detection. *CoRR* abs/1703.02716.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *ICCV*, 5783–5792.

Yang, K.; Qiao, P.; Li, D.; Lv, S.; and Dou, Y. 2018. Exploring Temporal Preservation Networks for Precise Temporal Action Localization. In *AAAI*, 7477–7484.

Yeung, S.; Russakovsky, O.; Mori, G.; and Feifei, L. 2016. End-to-End Learning of Action Detection from Frame Glimpses in Videos. In *CVPR*, 2678–2687.

Yuan, J.; Ni, B.; Yang, X.; and Kassim, A. A. 2016. Temporal Action Localization with Pyramid of Score Distribution Features. In *CVPR*, 3093–3102.

Yuan, Z.; Stroud, J. C.; Lu, T.; and Deng, J. 2017. Temporal Action Localization by Structured Maximal Sums. In *CVPR*, 3684–3692.

Zhang, Z.; Zou, Q.; Wang, Q.; Lin, Y.; and Li, Q. 2018. Instance Similarity Deep Hashing for Multi-Label Image Retrieval. *CoRR* abs/1803.02987.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal Action Detection with Structured Segment Networks. In *ICCV*, 2933–2942.

Zhou, Z. H.; Zhang, M. L.; Huang, S. J.; and Li, Y. F. 2008. Multi-Instance Multi-Label Learning. *AI* 176(1):2291–2320.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*, 2921–2929.