A Hybrid Attention Mechanism for Weakly-Supervised Temporal Action Localization

Ashraful Islam[1], **Chengjiang Long** [2], **Richard Radke** [1]
[1] Rensselaer Polytechnic Institute
[2] JD Digits AI Lab
islama6@rpi.edu, chengjiang.long@jd.com, rjradke@ecse.rpi.edu

Abstract

Weakly supervised temporal action localization is a challenging vision task due to the absence of ground-truth temporal locations of actions in the training videos. With only video-level supervision during training, most existing methods rely on a Multiple Instance Learning (MIL) framework to predict the start and end frame of each action category in a video. However, the existing MIL-based approach has a major limitation of only capturing the most discriminative frames of an action, ignoring the full extent of an activity. Moreover, these methods cannot model background activity effectively, which plays an important role in localizing foreground activities. In this paper, we present a novel framework named HAM-Net with a hybrid attention mechanism which includes temporal soft, semi-soft and hard attentions to address these issues. Our temporal soft attention module, guided by an auxiliary background class in the classification module, models the background activity by introducing an "action-ness" score for each video snippet. Moreover, our temporal semi-soft and hard attention modules, calculating two attention scores for each video snippet, help to focus on the less discriminative frames of an action to capture the full action boundary. Our proposed approach outperforms recent state-of-the-art methods by at least 2.2% mAP at IoU threshold 0.5 on the THUMOS14 dataset, and by at least 1.3% mAP at IoU threshold 0.75 on the ActivityNet1.2 dataset.

Introduction

Temporal action localization refers to the task of predicting the start and end times of all action instances in a video. There has been remarkable progress in fully-supervised temporal action localization (Tran et al. 2020; Zhao et al. 2017; Chao et al. 2018; Lin et al. 2018; Xu et al. 2020). However, annotating the precise temporal ranges of all action instances in a video dataset is expensive, time-consuming, and error-prone. On the contrary, weakly supervised temporal action localization (WTAL) can greatly simplify the data collection and annotation cost.

WTAL aims at localizing and classifying all action instances in a video given only video-level category label during training stage. Most existing WTAL methods rely on the multiple instance learning (MIL) paradigm (Paul, Roy, and Roy-Chowdhury 2018; Liu, Jiang, and Wang 2019; Islam and Radke 2020). In this paradigm, a video consists of several snippets; snippet-level class scores, commonly known as Class Activation Sequences (CAS), are calculated and then temporally pooled to obtain video-level class scores. The action proposals are generated by thresholding the snippet-level class scores. However, this framework has a major issue: it does not necessarily capture the full extent of an action instance. As training is performed to minimize the video-level classification loss, the network predicts higher CAS values for the discriminative parts of actions, ignoring the less discriminative parts. For example, an action might consist of several sub-actions (Hou, Sukthankar, and Shah 2017). In the MIL paradigm, only a particular sub-action might be detected, ignoring the other parts of the action.

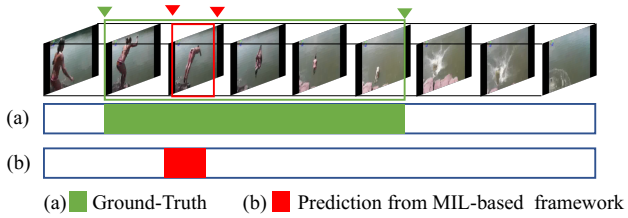


Figure 1: The existing MIL framework does not necessarily capture the full extent of an action instance. In this example of a diving activity. (a) shows the ground-truth localization, and (b) shows the prediction from an MIL-based WTAL framework. The MIL framework only captures the most discriminative part of the diving activity, ignoring the beginning and ending parts of the full action.

An illustrative example of a diving activity is presented in Figure 1. We observe that only the most discriminative location of the full diving activity is captured by the MIL framework. Capturing only the most distinctive part of an action is sufficient to produce a high video-level classification accuracy, but does not necessarily result in good temporal localization performance. Another issue with the existing framework is modeling the background activity effectively so that background frames are not included in the temporal localization prediction. It has been shown previously that background activity plays an important role in action localization

(Lee, Uh, and Byun 2020). Without differentiating the background frames from the foreground ones, the network might include the background frames to minimize foreground classification loss, resulting in many false positive localization predictions.

In this paper, we propose a new WTAL framework named HAM-Net with a hybrid attention mechanism to solve the above-mentioned issues. Attention mechanism has been used in successfully used in deep learning (Islam et al. 2020; Vaswani et al. 2017; Shi et al. 2020). HAM-Net produces soft, semi-soft and hard attentions to detect the full temporal span of action instances and to model background activity, as illustrated in Figure 2.

Our framework consists of (1) a classification branch that predicts class activation scores for all action instances including background activity, and (2) an attention branch that predicts the "action-ness" scores of a video snippet. The snippet-level class activation scores are also modulated by three snippet-level attention scores, and temporally pooled to produce video-level class scores.

To capture the full action instance, we drop the more discriminative parts of the video, and focus on the less discriminative parts. We achieve this by calculating semi-soft attention scores and hard attention scores for all snippets in the video. The semi-soft attention scores drop the more discriminative portions of the video by assigning zero value to the snippets that have soft-attention score greater than a threshold, and the scores for the other portions remain the same as the soft-attention scores. The video-level classification scores guided by the semi-soft attentions contain only foreground classes. On the other hand, the hard-attention score drops the more discriminative parts of the video, and assigns the attention scores of the less discriminative parts to one, which ensures that video-level class scores guided by this hard attention contain both foreground and background classes. Both the semi-soft and hard attentions encourage the model to learn the full temporal boundary of an action in the video.

To summarize, our contributions are threefold: (1) we propose a novel framework with a hybrid attention mechanism to model an action in its entirety; (2) we present a background modeling strategy by attention scores guided using an auxiliary background class; and (3) we achieve state-of-the-art performance on both the THUMOS14 (Jiang et al. 2014) and ActivityNet (Caba Heilbron et al. 2015) datasets. Specifically, we outperform state-of-the-art methods by 2.2% mAP at IoU threshold 0.5 on the THUMOS14 dataset, and 1.3% mAP at IoU threshold 0.75 on the ActivityNet1.2 dataset.

Related Work

Action Analysis with Full Supervision Due to the representation capability of deep learning based models, and the availability of large scale datasets (Jiang et al. 2014; Caba Heilbron et al. 2015; Sigurdsson et al. 2016; Gu et al. 2018; Kay et al. 2017), significant progress has been made in the domain of video action recognition. To design motion cues, the two-stream network (Simonyan and Zisserman 2014) incorporated optical flow (Horn and Schunck 1981) as a separate stream along with RGB frames. 3D convolutional networks have demonstrated better representations for video (Carreira and Zisserman 2017; Tran et al. 2015, 2020). For fully-supervised temporal action localization, several recent methods adopt a two-stage strategy (Tran et al. 2020; Zhao et al. 2017; Chao et al. 2018; Lin et al. 2018).

Weakly Supervised Temporal Action Localization In terms of exisiting WTAL methods, UntrimmedNets (Wang et al. 2017) introduced a classification module for predicting a classification score for each snippet, and a selection module to select relevant video segments. On top of that, STPN (Nguyen et al. 2018) added a sparsity loss and class-specific proposals. AutoLoc (Shou et al. 2018) introduced the outer-inner contrastive loss to effectively predict the temporal boundaries. W-TALC (Paul, Roy, and Roy-Chowdhury 2018) and Islam and Radke (Islam and Radke 2020) incorporated distance metric learning strategies.

MAAN (Yuan et al. 2019) proposed a novel marginalized average aggregation module and latent discriminative probabilities to reduce the difference between the most salient regions and the others. TSM (Yu et al. 2019) modeled each action instance as a multi-phase process to effectively characterize action instances. WSGN (Fernando, Tan, and Bilen 2020) assigns a weight to each frame prediction based on both local and global statistics. DGAM (Shi et al. 2020) used a conditional Variational Auto-Encoder (VAE) to separate the attention, action, and non-action frames. CleanNet (Liu et al. 2019) introduced an action proposal evaluator that provides pseudo-supervision by leveraging the temporal contrast in snippets. 3C-Net (Narayan et al. 2019) adopted three loss terms to ensure the separability, to enhance discriminability, and to delineate adjacent action sequences. Moreover, BaS-Net (Lee, Uh, and Byun 2020) and Nguyen *et al*(Nguyen, Ramanan, and Fowlkes 2019) modeled background activity by introducing an auxiliary background class. However, none of these approaches explicitly resolve the issue of modeling an action instance in its entirety.

To model action completeness, Hide-and-Seek (Singh and Lee 2017) hid part of the video to discover other relevant parts, and Liu *et al*(Liu, Jiang, and Wang 2019) proposed a multi-branch network where each branch predicts distinctive action parts. Our approach has similar motivation, but differs in that we hide the most discriminative parts of the video instead of random parts.

Proposed Method

Problem Formulation

Assume a training video V containing activity instances chosen from n_c activity classes. A particular activity can occur in the video multiple times. Only the video-level action instances are given. Denote the video-level activity instances as $\mathbf{y} \in \{0,1\}^{n_c}$, where $y_j = 1$ only if there is at least one instance of the j-th action class in the video, and $y_j = 0$ if there is no instance of the j-th activity. Note that neither the frequency nor the order of the action instances in the video is provided. Our goal is to create a model that is trained only

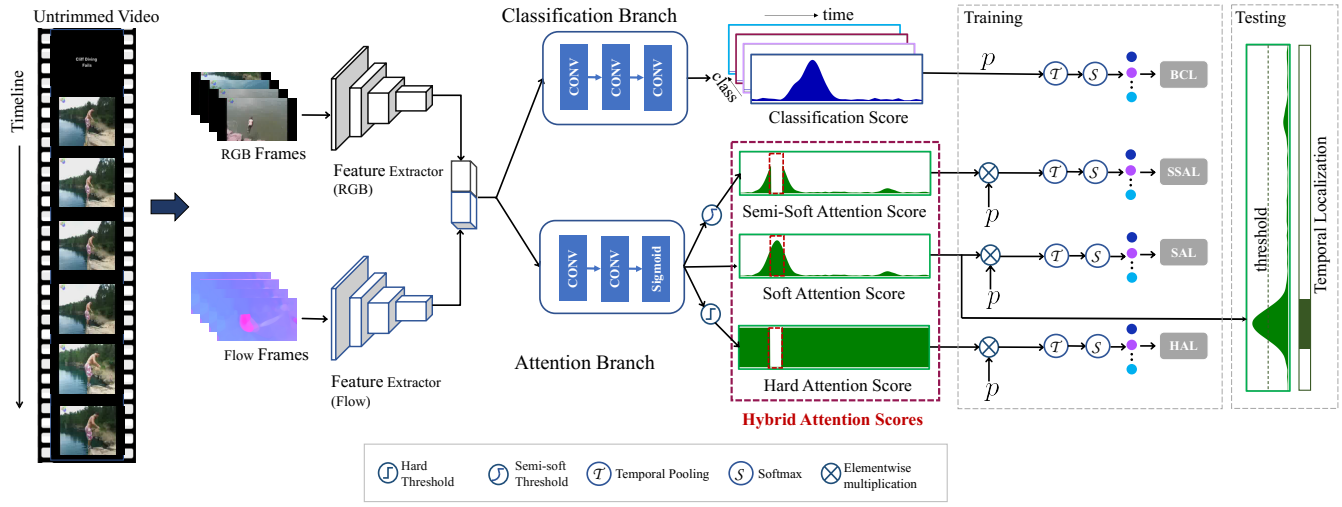


Figure 2: Overview of our proposed framework HAM-Net. Snippet-level features of both RGB and flow frames are extracted and separately fed into a classification branch and an attention branch with a hybrid attention mechanism. Three attention scores are calculated: soft attention, semi-soft attention, and hard attention, which are multiplied with snippet-level classification scores to obtain attention-guided class scores. The network is trained using four attention-guided losses: base classification loss (BCL), soft attention loss (SAL), semi-soft attention loss (SSAL), and hard attention loss (HAL), as well as sparsity loss and guide loss.

with video-level action classes, and predicts temporal location of activity instances during evaluation, *i.e.*, for a testing video it outputs a set of tuples (t_s, t_e, ψ, c) where t_s and t_e are the start and end frames of an action, c is the action label, and ψ is the activity score.

Snippet-Level Classification

In our proposed HAM-Net, as illustrated in Figure 2, for each video, we first divide it into non-overlapping snippets to extract snippet-level features. Using a snippet-level representation rather than a frame-level representation allows us to use existing 3D convolutional features extractors that can effectively model temporal dependencies in the video. Following the two-stream strategy (Carreira and Zisserman 2017; Feichtenhofer, Pinz, and Zisserman 2016) for action recognition, we extract snippet-level features for both the RGB and flow streams, denoted as $\mathbf{x}_i^{\text{RGB}} \in \mathbb{R}^D$ and $\mathbf{x}_i^{\text{Flow}} \in \mathbb{R}^D$ respectively. We concatenate both streams to obtain full snippet features $\mathbf{x}_i \in \mathbb{R}^{2D}$ for the i-th snippet, resulting in a high-level representation of the snippet feature that contains both appearance and motion cues.

To determine the temporal locations of all activities in the video, we calculate the snippet-level classification scores from classification branch, which is a convolutional neural network that outputs the class logits commonly known as Class Activation Sequences (CAS) (Shou et al. 2018). We denote the snippet level CAS for all classes for the i-th snippet as $\mathbf{s}_i \in \mathbb{R}^{c+1}$. Here, the $c+1$-th class is the background class. Since we only have the video-level class scores as ground truth, we need to pool the snippet-level scores $\mathbf{s}_i$ to obtain video-level class scores. There are several pooling strategies in the literature to obtain video-level scores from snippet level scores. We adopt the top-k strategy (Islam and Radke 2020; Paul, Roy, and Roy-Chowdhury 2018) in our setting. Specifically, the temporal pooling is performed by aggregating the top-k values from the temporal dimension for each class:

$$v_j = \max_{\substack{l \subset \{1,2,\ldots,T\} \\ |l|=k}} \frac{1}{k} \sum_{i \in l} \mathbf{s}_i(j) \tag{1}$$

Next, we calculate the video-level class scores by applying softmax operations along the class dimension:

$$p_j = \frac{\exp(v_j)}{\sum_{j'=1}^{c+1} \exp(v'_j)} \tag{2}$$

where $j = 1, 2, \ldots, c+1$.

The base classification loss is calculated as the cross entropy loss between the ground-truth video-level class scores $\mathbf{y}$ and the predicted scores $\mathbf{p}$:

$$\mathcal{L}_{\text{BCL}} = -\sum_{j=1}^{c+1} y_j \log(p_j) \tag{3}$$

Note that every untrimmed video contains some background portions where there are no actions occurring. These background portions are modeled as a separate class in the classification branch. Hence, the ground-truth background class $y_{c+1} = 1$ in Eqn. 3. One major issue of this approach is that there are no negative samples for the background class, and the model cannot learn background activity by only optimizing with positive samples. To overcome this issue, we propose a hybrid attention mechanism in the attention branch to further explore the "action-ness" score of each segment.

A Hybrid Attention Mechanism for Weak Supervision

To suppress background classes from the video, we incorporate an attention module to differentiate foreground and background actions following the background modeling strategy in several weakly-supervised action detection papers (Nguyen, Ramanan, and Fowlkes 2019; Lee, Uh, and Byun 2020; Liu et al. 2019). The goal is to predict an attention score for each snippet that is lower in the frames where there is no activity instance (i.e., background activity) and higher for other regions. Although the classification branch predicts the probability of background action in the snippets, a separate attention module is more effective to differentiate between the foreground and background classes for several reasons. First, most of the actions in a video occur in regions where there are high motion cues; the attention branch can initially detect the background region only from motion features. Second, it is easier for a network to learn two classes (foreground vs. background) rather than a large number of classes with weak supervision.

Soft Attention Score The input to the attention module is the snippet-level feature x_i, and it returns a single foreground attention score a_i:

$$a_i = g(\mathbf{x}_i; \Theta), \tag{4}$$

where $a_i \in [0, 1]$, and $g(\cdot; \Theta)$ is a function with parameters Θ that is designed with two temporal convolution layers followed by a sigmoid activation layer.

To create negative samples for the background class, we multiply the snippet level class logit (i.e., CAS) $s_i(j)$ for each class j with the snippet-level attention score a_i for the i-th snippet, and obtain attention-guided snippet-level class scores $s_i^{\text{attn}}(j) = s_i(j) \otimes a_i$, where $\otimes$ is the element-wise product. s^{attn} serves as a set of snippets without any background activity, which can be considered as negative samples for the background class. Following Eqns. 1 and 2, we obtain video level attention-guided class scores p_j^{attn} for class label j:

$$v_j^{\text{attn}} = \max_{\substack{l \subset \{1,2,\ldots,T\} \\ |l|=k}} \frac{1}{k} \sum_{i \in l} \mathbf{s}_i^{\text{attn}}(j) \tag{5}$$

$$p_j^{\text{attn}} = \frac{\exp(v_j^{\text{attn}})}{\sum_{j'=1}^{c+1} \exp(v_{j'}^{\text{attn}})} \tag{6}$$

where $j = 1, 2, \ldots, c+1$. Note that p_j^{attn} does not contain any background class, since the background class has been suppressed by the attention score a_i. From p_j^{attn}, we calculate the soft attention-guided loss (SAL) function

$$\mathcal{L}_{\text{SAL}} = -\sum_{j=1}^{c+1} y_j^f \log(p_j) \tag{7}$$

Here, y_j^f contains only the foreground activities, i.e., the background class $y_{c+1}^f = 0$, since the attention score suppresses the background activity.

Semi-Soft Attention Score Given the snippet-level class score s_i and soft-attention score a_i for the i-th snippet, we calculate the semi-soft attention scores by thresholding the soft attention a_i by a particular value $\gamma \in [0, 1]$,

$$a_i^{\text{semi-soft}} = \begin{cases} a_i, & \text{if } a_i < \gamma \\ 0, & \text{otherwise} \end{cases}$$

Note that the semi-soft attention $a_i^{\text{semi-soft}}$ both drops the most discriminative regions and attends to the foreground snippets only; hence, the semi-soft attention guided video-level class scores will only contain foreground activities. This design helps to better model the background, as discussed in the ablation studies section. Denote the video-level class scores associated with semi-soft attention as $p_j^{\text{semi-soft}}$, where $j = 1, 2, \ldots, c+1$. We calculate the semi-soft attention loss:

$$\mathcal{L}_{\text{SSAL}} = -\sum_{j=1}^{c+1} y_j^f \log(p_j^{\text{semi-soft}}) \tag{8}$$

where y^f is the ground truth label without background activity, i.e., $y_{c+1}^f = 0$, since the semi-soft attention suppresses the background snippets along with removing the most discriminative regions.

Hard Attention Score In contrast to semi-soft attention, hard attention score is calculated by

$$a_i^{\text{hard}} = \begin{cases} 1, & \text{if } a_i < \gamma \\ 0, & \text{otherwise} \end{cases}$$

With hard attention score, we obtain another set of video-level class scores by multiplying them with the original snippet-level logit $s_i(j)$ and temporally pooling the scores following Eqn. 1 and Eqn. 2. We obtain the hard-attention loss:

$$\mathcal{L}_{\text{HAL}} = -\sum_{j=1}^{c+1} y_j \log(p_j^{\text{hard}}) \tag{9}$$

where y is the ground truth label with background activity, i.e., $y_{c+1} = 1$, since the hard attention does not suppress the background snippets, rather, it only removes the more discriminative regions of a video.

Loss Functions Finally, we train our proposed HAM-Net using the following joint loss function:

$$\begin{aligned} \mathcal{L} = \lambda_0 \mathcal{L}_{\text{BCL}} + \lambda_1 \mathcal{L}_{\text{SAL}} + \lambda_2 \mathcal{L}_{\text{SSAL}} + \lambda_3 \mathcal{L}_{\text{HAL}} \\ + \alpha \mathcal{L}_{\text{sparse}} + \beta \mathcal{L}_{\text{guide}} \end{aligned} \tag{10}$$

where $\mathcal{L}_{\text{sparse}}$ is sparse loss, $\mathcal{L}_{\text{guide}}$ is guide loss, and λ_0, λ_1, λ_2, λ_3, α, and β are hyper-parameters.

The sparsity loss $\mathcal{L}_{\text{sparse}}$ is based on the assumption that an action is recognizable from a sparse subset of the video segments (Nguyen et al. 2018). The sparsity loss is calculated as the L1-norm of the soft-attention scores:

$$\mathcal{L}_{\text{sparse}} = \sum_{i=1}^{T} |a_i| \tag{11}$$

Regarding the guide loss $\mathcal{L}_{\text{guide}}$, we consider the soft-attention score a_i as a form of binary classification score for each snippet, where there are only two classes, foreground and background, the probabilities of which are captured by a_i and $1 - a_i$. Hence, $1 - a_i$ can be considered as the probability of the i-th snippet containing background activity. On the other hand, the background class is also captured by the class activation logits $\mathbf{s}_i(\cdot) \in \mathbb{R}^{c+1}$. To guide the background class activation to follow the background attention, we first calculate the probability of a particular segment being background activity,

$$\bar{s}_{c+1} = \frac{\exp(s_{c+1})}{\sum_{j=1}^{c} \exp(s_j)} \tag{12}$$

and then add a guide loss so that the absolute difference between the background class probability and the background attention is minimized:

$$\mathcal{L}_{\text{guide}} = \sum_{i=1}^{T} |1 - a_i - \bar{s}_{c+1}| \tag{13}$$

Temporal Action Localization

For temporal localization, we first discard classes which have video-level class score less than a particular threshold (set to 0.1 in our experiments). For the remaining classes, we first discard the background snippets by thresholding the soft attention scores a_i for all snippets i, and obtain class-agnostic action proposals by selecting the one-dimensional connected components of the remaining snippets. Denote the candidate action locations as $\{(t_s, t_e, \psi, c)\}$, where t_s is the start time, t_e is the end time, and ψ is the classification score for class c. We calculate the classification score following the outer-inner score of AutoLoc (Shou et al. 2018). Note that for calculating class-specific scores, we use the attention-guided class logits s_c^{attn},

$$\psi = \psi_{\text{inner}} - \psi_{\text{outer}} + \zeta p_c^{\text{attn}} \tag{14}$$

$$\psi_{\text{inner}} = \text{Avg}(s_c^{\text{attn}}(t_s : t_e)) \tag{15}$$

$$\psi_{\text{outer}} = \text{Avg}(s_c^{\text{attn}}(t_s - l_m : t_s) + s_c^{\text{attn}}(t_e : t_e + l_m)) \tag{16}$$

where ζ is a hyper-parameter, $l_m = (t_e - t_s)/4$, p_c^{attn} is the video-level score for class c, and $s_c^{\text{attn}}(\cdot)$ is the snippet-level class logit for class c. We apply different thresholds for obtaining action proposals, and remove the overlapping segments with non-maximum suppression.

Experiments

Experimental Settings

Datasets We evaluate our approach on two popular action localization datasets: THUMOS14 (Jiang et al. 2014) and ActivityNet1.2 (Caba Heilbron et al. 2015). **THUMOS14** contains 200 validation videos for training and 213 testing videos for testing with 20 action categories. This is a challenging dataset with around 15.5 activity segments and 71% background activity per video. **ActivityNet1.2** dataset contains 4,819 videos for training and 2,382 videos for testing with 200 action classes. It contains around 1.5 activity instances (10 times sparser than THUMOS14) and 36% background activity per video.

Evaluation Metrics For evaluation, we use the standard protocol and report mean Average Precision (mAP) at various intersection over union (IoU) thresholds. The evaluation code provided by ActivityNet (Caba Heilbron et al. 2015) is used to calculate the evaluation metrics.

Implementation Details For feature extraction, we sample the video streams into non-overlapping 16 frame chunks for both the RGB and the flow stream. Flow streams are created using the TV-L1 algorithm (Wedel et al. 2008). We use the I3D network (Carreira and Zisserman 2017) pre-trained on the Kinetics dataset (Kay et al. 2017) to extract both RGB and flow features, and concatenate them to obtain 2048-dimensional snippet-level features. During training we randomly sample 500 snippets for THUMOS14 and 80 snippets for ActivityNet, and during evaluation we take all the snippets. The classification branch is designed as two temporal convolution layers with kernel size 3, each followed by LeakyReLU activation, and a final linear fully-connected layer for predicting class logits. The attention branch consists of two temporal convolution layers with kernel size 3 followed by a sigmoid layer to predict attention scores between 0 and 1.

We use the Adam (Kingma and Ba 2015) optimizer with learning rate 0.00001, and train for 100 epochs for THUMOS14 and 20 epochs for ActivityNet. For THUMOS14, we set $\lambda_0 = \lambda_1 = 0.8$, $\lambda_2 = \lambda_3 = 0.2$, $\alpha = \beta = 0.8$, $\gamma = 0.2$, and $k = 50$ for top-k temporal pooling. For ActivityNet, we set $\alpha = 0.5$, $\beta = 0.1$, $\lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = 0.5$, and $k = 4$, and apply additional average pooling to post-process the final CAS. All the hyperparameters are determined from grid search. For action localization, we set the thresholds from 0.1 to 0.9 with a step of 0.05, and perform non-maximum suppression to remove overlapping segments.

Ablation Studies

We conduct a set of ablation studies on the THUMOS14 dataset to analyze the performance contribution of each component of our proposed HAM-Net. Table 1 shows the performance of our method with respect to different loss terms. We use "AVG mAP" for the performance metric, which is the average of mAP values for different IoU thresholds (0.1:0.1:0.7). The first five experiments are trained without SSAL or HAL loss, i.e., without any temporal dropping mechanism, which we denote as "MIL-only mode", and the remaining experiments trained with those losses are denoted as "MIL and Drop mode". Figure 3 shows the localization prediction of different experiments on a representative video. Our analysis shows that all the loss components are required to achieve the maximum performance.

Importance of sparsity and guide loss Table 1 shows that both sparsity and guide loss are important to acheive better performance. Specifically, in "MIL-only mode", adding both sparsity and guide loss provides 4% mAP gain, and in "MIL and Drop mode", the mAP gain is 9%, suggesting that these losses are more important in "MIL and Drop mode". Note that for SSAL and HAL, the discriminativeness of a snippet is measured by the soft-attention scores that are learned by

$\mathcal{L}_{BCL}$	$\mathcal{L}_{SAL}$	$\mathcal{L}_{HAL}$	$\mathcal{L}_{SSAL}$	$\mathcal{L}_{sparse}$	$\mathcal{L}_{guide}$	AVG mAP
1) ✓	-	-	-	-	-	24.6
2) ✓	✓	-	-	-	-	30.8
3) ✓	✓	-	-	-	✓	28.9
4) ✓	✓	-	-	✓	-	30.9
5) ✓	✓	-	-	✓	✓	**34.8**
6) ✓	✓	✓	✓	-	-	30.9
7) ✓	✓	✓	✓	-	✓	31.1
8) ✓	✓	✓	✓	✓	-	37.9
9) ✓	✓	✓	-	✓	✓	36.6
10) ✓	✓	-	✓	✓	✓	38.1
11) ✓	✓	✓	✓	✓	✓	**39.8**

Table 1: Ablation study on the effectiveness of different combination of loss functions in the localization performance on THUMOS14 in terms of mAP. Here, AVG mAP means the average of mAP values from IoU thresholds 0.1 to 0.7. Adding $\mathcal{L}_{AL}$ in the total loss function improves the mAP from 34.8 to 39.8.

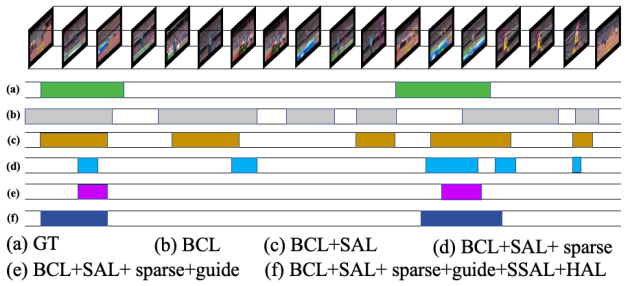


Figure 3: Visualization of the effects of different loss functions on the final localization for a video containing the Long Jump activity. (a) is the ground-truth action location. (b) represents only MIL loss, which predicts many false positives. After adding sparsity and guide loss, in (d) we get rid of those false positives, but still cannot capture full temporal boundaries. (e) shows results from our approach which captures full action boundaries.

sparsity and guide loss. Without sparsity loss, the majority of the soft-attention scores remain close to 1, making the snippet dropping strategy ineffective. Moreover, the guide loss itself does not increase the localization performance significantly without the sparsity loss (experiment 3 and experiment 7 in Table 1); however, combined with sparsity loss it shows the best performance improvement (experiment 5 and experiment 11 in Table 1).

Importance of attention losses We observe that the attention losses can significantly improve the performance. Table 1 shows that only incorporating $\mathcal{L}_{SAL}$ achieves 6.2% average mAP gain over the BCL-only model. From experiment-9 and experiment-10 in Table 1, we see that both HAL and SSAL individually improve the performance, and we get the best performance when we combine them. Specifically, the combination of HAL and SSAL improves the performance by 5% over the best score in "MIL-only mode". Figure 5 shows visualization examples of the effectiveness of the losses on a representative video. We can observe that the MIL-only model fails to capture several parts of a full action instance (i.e. , Long Jump). Incorporating attention losses helps to capture the action in its entirety.

Importance of dropping snippets by selective thresholding For calculating the HAL loss and the SSAL loss, we drop the more discriminative parts of the video and train on the less discriminative parts, assuming that focusing on less discriminative parts will help the model to learn the completeness of actions. To confirm our assumption here, we create two baselines: "**ours with random drop**" where we randomly drop video snippets, similar to Hide-and-Seek (Singh and Lee 2017), and "**ours with inverse drop**", where we drop the less discriminative parts instead of dropping the most discriminative parts. We show the performance comparison between these models in Figure 4a. Results show that randomly dropping snippets is slightly more effective than the baseline, and dropping the less discriminative parts decreases the localization performance. Our approach performs much better than randomly dropping snippets or dropping less discriminative snippets, which proves the efficacy of selectively dropping more discriminative foreground snippets.

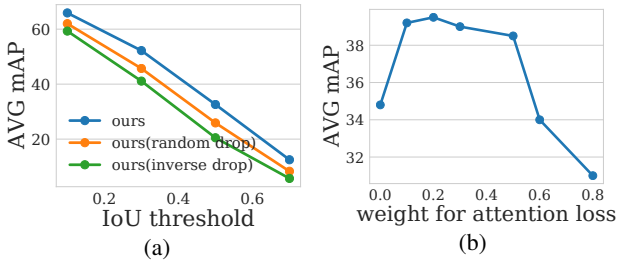


Figure 4: (a) Ablation study on the importance of dropping snippets by selective thresholding. Other approaches like random dropping or inverse selective thresholding do not work well. (b) Ablation study on the importance of SSAL and HAL. A lower weight causes the model to learn only the most distinctive parts, and a higher weight gives too much focus to the less distinctive parts.

Ablation on λ_2 and λ_3 For this analysis, we set $\lambda_2 = \lambda_3 = \lambda$. In Figure 4b, we analyze the effect of λ to the performance. Note that $\lambda = 0$ denotes "MIL only mode", which achieves an average mAP of 34.8%. Increasing the value of λ results in performance improvement until λ reaches 0.2, after which we observe performance degradation. The reason is that a lower weight does not incorporate $\mathcal{L}_{SSAL}$ and $\mathcal{L}_{HAL}$ effectively during training. On the contrary, a higher weight provides too much importance on the less discriminative parts, which might cause the model to ignore the more discriminative regions in every iteration, resulting in poor localization performance. The optimum value of 0.2 balances out both of the issues.

Performance Comparison to State-of-the-Art

Table 2 summarizes performance comparisons between our proposed HAM-Net and state-of-the-art fully-supervised and weakly-supervised TAL methods on the THUMOS14

Method	Feature	IoU							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
R-C3D (Xu, Das, and Saenko 2017)	-	54.5	51.5	44.8	35.6	28.9	-	-	-
SSN (Zhao et al. 2017)	-	66.0	59.4	51.9	41.0	29.8	-	-	-
BSN (Lin et al. 2018)	-	-	-	53.5	45.0	36.9	28.4	20.0	-
G-TAD (Xu et al. 2020)	-	-	-	54.5	47.6	40.2	**30.8**	**23.4**	-
P-GCN (Zeng et al. 2019)	-	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**	-	-	-
Hide-and-Seek (Singh and Lee 2017)	-	36.4	27.8	19.5	12.7	6.8	-	-	-
UntrimmedNets (Wang et al. 2017)	-	44.4	37.7	28.2	21.1	13.7	-	-	-
STPN (Nguyen et al. 2018)	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	26.4
AutoLoc (Shou et al. 2018)	UNT	-	-	35.8	29.0	21.2	13.4	5.8	-
W-TALC (Paul, Roy, and Roy-Chowdhury 2018)	I3D	55.2	49.6	40.1	31.1	22.8	-	7.6	-
Liu *et al* (Liu, Jiang, and Wang 2019)	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.4
MAAN (Yuan et al. 2019)	I3D	59.8	50.8	41.1	30.6	20.3	12.0	6.9	31.6
TSM (Yu et al. 2019)	I3D	-	-	39.5	-	24.5	-	7.1	-
CleanNet (Liu et al. 2019)	UNT	-	-	37.0	30.9	23.9	13.9	7.1	-
3C-Net (Narayan et al. 2019)	I3D	56.8	49.8	40.9	32.3	24.6	-	7.7	-
Nguyen *et al* (Nguyen, Ramanan, and Fowlkes 2019)	I3D	60.4	56.0	46.6	37.5	26.8	17.6	9.0	36.3
WSGN (Fernando, Tan, and Bilen 2020)	I3D	57.9	51.2	42.0	33.1	25.1	16.7	8.9	33.6
Islam *et al* (Islam and Radke 2020)	I3D	62.3	-	46.8	-	29.6	-	9.7	-
BaS-Net (Lee, Uh, and Byun 2020)	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.3
DGAM (Shi et al. 2020)	I3D	60.0	54.2	46.8	38.2	28.8	19.8	11.4	37.0
HAM-Net (Ours)	I3D	**65.4**	**59.0**	**50.3**	**41.1**	**31.0**	**20.7**	**11.14**	**39.8**

Table 2: Comparison of our algorithm with other state-of-the-art methods on the THUMOS14 dataset for temporal action localization.

dataset. We report mAP scores at different IoU thresholds. 'AVG' is the average mAP for IoU 0.1 to 0.7 with step size of 0.1. With weak supervision, our proposed HAM-Net achieves state-of-the-art scores on all IoU thresholds. Specifically, HAM-Net achieves 2.2% more mAP than the current best scores at IoU threshold 0.5. Moreover, our HAM-Net outperforms some fully-supervised TAL models, and even shows comparable results with some recent fully-supervised TAL methods.

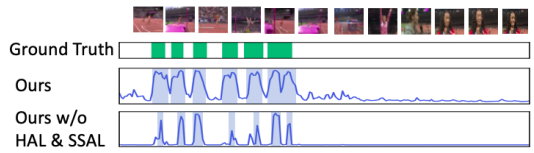
In Table 3, we evaluate HAM-Net on the ActivityNet1.2 dataset. HAM-Net outperforms other WTAL approaches on ActivityNet1.2 across all metrics, verifying the effectiveness of our proposed HAM-Net.

	Method	IoU			
		0.5	0.75	0.95	AVG
Full	SSN	41.3	27.0	6.1	26.6
Weak	UntrimmedNets	7.4	3.2	0.7	3.6
	AutoLoc	27.3	15.1	3.3	16.0
	W-TALC	37.0	12.7	1.5	18.0
	Islam *et al*	35.2	-	-	-
	TSM	28.3	17.0	3.5	17.1
	3C-Net	35.4	-	-	21.1
	CleanNet	37.1	20.3	5.0	21.6
	Liu *et al*	36.8	22.0	5.6	22.4
	BaS-Net	34.5	22.5	4.9	22.2
	DGAM	**41.0**	23.5	**5.3**	24.4
	HAM-Net (Ours)	**41.0**	**24.8**	**5.3**	**25.1**

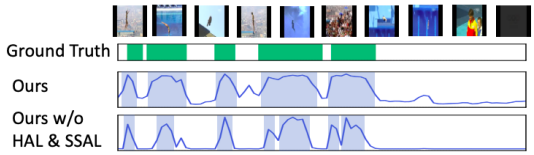
Table 3: Comparison of our algorithm with other state-of-the-art methods on the ActivityNet1.2 validation set for temporal action localization. AVG means average mAP from IoU 0.5 to 0.95 with 0.05 increment.

Qualitative Performance

We show some representative examples in Fig. 5. For each video, the top row shows example frames, the next row represents ground-truth localization, "Ours" is our prediction, and "Ours w/o HAL & SSAL" is ours trained without $\mathcal{L}_{\text{HAL}}$ and $\mathcal{L}_{\text{SSAL}}$. Fig. 5 shows that our model clearly captures the full temporal extent of activities, while "ours w/o HAL & SSAL" focuses only on the more discriminative snippets.



(a) High Jump



(b) Diving

Figure 5: Qualitative results on THUMOS14. The horizontal axis denotes time. On the vertical axis, we sequentially plot the ground truth detection, our detection scores, and detection scores of ours without HAL and SSAL. SSAL and HAL helps to learn the full context of an action.

Conclusion

We presented a novel framework called HAM-Net to learn temporal action localization from only video-level supervision during training. We introduced a hybrid attention mechanism including soft, semi-soft, and hard attentions to differentiate background frames from foreground ones and to capture the full temporal boundaries of the actions in the video, respectively. We perform extensive analysis to show the effectiveness of our approach. Our approach achieves state-of-the-art performance on both the THUMOS14 and ActivityNet1.2 dataset.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 961–970.

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (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. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 1130–1139.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional Two-Stream Network Fusion for Video Action Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1933–1941.

Fernando, B.; Tan, C.; and Bilen, H. 2020. Weakly supervised Gaussian networks for action detection. In *The IEEE Winter Conference on Applications of Computer Vision*, 537–546.

Gu, C.; Sun, C.; Vijayanarasimhan, S.; Pantofaru, C.; Ross, D. A.; Toderici, G.; Li, Y.; Ricco, S.; Sukthankar, R.; Schmid, C.; and Malik, J. 2018. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 6047–6056.

Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, 319–331. International Society for Optics and Photonics.

Hou, R.; Sukthankar, R.; and Shah, M. 2017. Real-Time Temporal Action Localization in Untrimmed Videos by Sub-Action Discovery. In *BMVC*, volume 2, 7.

Islam, A.; Long, C.; Basharat, A.; and Hoogs, A. 2020. DOA-GAN: Dual-Order Attentive Generative Adversarial Network for Image Copy-Move Forgery Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1133–1141.

Islam, A.; and Radke, R. 2020. Weakly Supervised Temporal Action Localization Using Deep Metric Learning. In *The IEEE Winter Conference on Applications of Computer Vision*, 547–556.

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. URL http://crcv.ucf.edu/THUMOS14. Last accessed on 02/12/2021.

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

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* .

Lee, P.; Uh, Y.; and Byun, H. 2020. Background Suppression Network for Weakly-supervised Temporal Action Localization. In *AAAI*, 1502–1511.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Liu, D.; Jiang, T.; and Wang, Y. 2019. Completeness Modeling and Context Separation for Weakly Supervised Temporal Action Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1298–1307.

Liu, Z.; Wang, L.; Zhang, Q.; Gao, Z.; Niu, Z.; Zheng, N.; and Hua, G. 2019. Weakly supervised temporal action localization through contrast based evaluation networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 3899–3908.

Narayan, S.; Cholakkal, H.; Khan, F. S.; and Shao, L. 2019. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 8679–8687.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6752–6761.

Nguyen, P. X.; Ramanan, D.; and Fowlkes, C. C. 2019. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, 5502–5511.

Paul, S.; Roy, S.; and Roy-Chowdhury, A. K. 2018. W-TALC: Weakly-supervised Temporal Activity Localization and Classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 563–579.

Shi, B.; Dai, Q.; Mu, Y.; and Wang, J. 2020. Weakly-Supervised Action Localization by Generative Attention Modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1735–1742.

Shou, Z.; Gao, H.; Zhang, L.; Miyazawa, K.; and Chang, S.-F. 2018. AutoLoc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 154–171.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision*, 2294–2303.

Simonyan, K.; and Zisserman, A. 2014. Two-stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, 568–576.

Singh, K. K.; and Lee, Y. J. 2017. Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-Supervised Object and Action Localization. *2017 IEEE International Conference on Computer Vision (ICCV)* 3544–3553.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. *2015 IEEE International Conference on Computer Vision (ICCV)* 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2020. A Closer Look at Spatiotemporal Convolutions for Action Recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 1510–1517.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. UntrimmedNets for weakly supervised action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4325–4334.

Wedel, A.; Pock, T.; Zach, C.; Bischof, H.; and Cremers, D. 2008. An Improved Algorithm for TV-L1 Optical Flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, 23–45.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. *2017 IEEE International Conference on Computer Vision (ICCV)* 5794–5803.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5783–5792.

Yu, T.; Ren, Z.; Li, Y.; Yan, E.; Xu, N.; and Yuan, J. 2019. Temporal structure mining for weakly supervised action detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 5522–5531.

Yuan, Y.; Lyu, Y.; Shen, X.; Tsang, I. W.; and Yeung, D.-Y. 2019. Marginalized Average Attentional Network for Weakly-Supervised Learning. In *International Conference on Learning Representations (ICLR)*.

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph Convolutional Networks for Temporal Action Localization. In *The IEEE International Conference on Computer Vision (ICCV)*, 1107–1116.

Zhao, Y. S.; Xiong, Y.; Wang, L.; Wu, Z.; Lin, D.; and Tang, X. 2017. Temporal Action Detection with Structured Segment Networks. *2017 IEEE International Conference on Computer Vision (ICCV)* 2933–2942.