arXiv:2008.07728v1 [cs.CV] 18 Aug 2020

Equivalent Classification Mapping for Weakly Supervised Temporal Action Localization

Le Yang[1], Dingwen Zhang[1,2], Tao Zhao[1], Junwei Han[1]
[1]Northwestern Polytechnical University, [2]Xidian University
{nwpuyangle, zdw2006yyy, taozhao2011, junweihan2010}@gmail.com

Abstract

Weakly supervised temporal action localization is a newly emerging yet widely studied topic in recent years. The existing methods can be categorized into two localization-by-classification pipelines, i.e., the pre-classification pipeline and the post-classification pipeline. The pre-classification pipeline first performs classification on each video snippet and then aggregate the snippet-level classification scores to obtain the video-level classification score, while the post-classification pipeline aggregates the snippet-level features first and then predicts the video-level classification score based on the aggregated feature. Although the classifiers in these two pipelines are used in different ways, the role they play is exactly the same—to classify the given features to identify the corresponding action categories. To this end, an ideal classifier can make both pipelines work. This inspires us to simultaneously learn these two pipelines in a unified framework to obtain an effective classifier. Specifically, in the proposed learning framework, we implement two parallel network streams to model the two localization-by-classification pipelines simultaneously and make the two network streams share the same classifier, thus achieving the novel Equivalent Classification Mapping (ECM) mechanism. Considering that an ideal classifier would make the classification results of the two network streams be identical and make the frame-level classification scores obtained from the pre-classification pipeline and the feature aggregation weights in the post-classification pipeline be consistent, we further introduce an equivalent classification loss and an equivalent weight transition module to endow the proposed learning framework with such properties. Comprehensive experiments are carried on three benchmarks and the proposed ECM achieves superior performance over other state-of-the-art methods.

1 Introduction

Temporal action localization aims to localize action instances from the given untrimmed video by determining the start temporal points, end temporal points, and the corresponding action categories. In the studied weakly supervised setting, the action localizers are learned directly from video-level labels, without requiring the fine segment-level annotations. Thus, weakly supervised temporal action localization can alleviate the burdensome and expensive human annotation and further realize the potential learning process on web-scale unlabeled videos.

In order to eliminate the ambiguity brought by the weak supervision, most previous works adopt the localization-by-classification pipelines to estimate the video-level classification score, which can be categorized into two main pipelines, i.e., the pre-classification pipeline (see Fig. 1 (a)) and the post-classification pipeline (see Fig. 1 (b)). Among the existing methods, UntrimmedNets [1] makes the pioneering exploration for the pre-classification pipeline. It first performs classification at each temporal point to obtain the *class activation sequence*, which is then aggregated to predict the

Preprint. Under review.

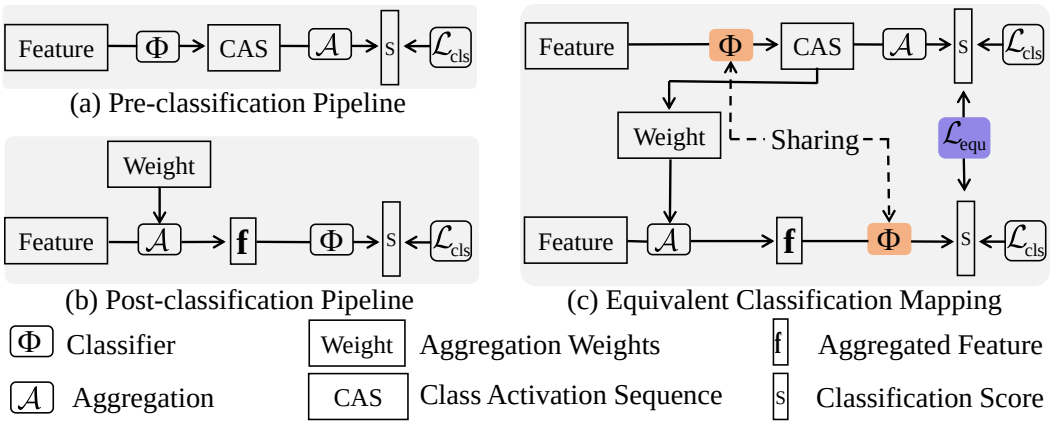

Figure 1: Illustration of different action localization pipelines. We propose to learn the identical classification mapping function from both the pre-classification pipeline and the post-classification pipeline, with the help of the basic classification loss $\mathcal{L}_{\text{cls}}$ and the equivalent classification loss $\mathcal{L}_{\text{equ}}$.

video-level classification score. Based on this work, CMCS [2] proposes a diversity loss to model the completeness of actions. Meanwhile 3C-Net [3] introduces action count cues to distinguish adjacent action sequences. In the aforementioned pre-classification pipelines, the classification mapping functions are learned based on each temporal point and the corresponding local neighbors. Such a mechanism would be beneficial to capturing local contrast information but tends to be less effective in perceiving long-term connections within each given video. There are also works, such as [4, 5, 6], which adopt the post-classification pipelines. These works first aggregate features from all temporal points to form the video-level feature representation, then they predict the video-level classification scores by classifying the aggregated feature representation. The advantage of these methods is that the aggregated feature can represent long-term relationships. However, when performing classification at each temporal point in the evaluation phase, the class activation sequences generated by these methods show insufficient discriminability to localize the actions that belong to different categories.

From the above discussion, we can observe that both the pre-classification pipeline and the post-classification pipeline aim at learning the effective classification mapping functions to predict the classification scores from the input features. The difference is that the pre-classification pipeline uses the classifier to perform classification on the features of each temporal point, while the post-classification pipeline uses the classifier to perform classification on the aggregated features of the whole video sequence. Note that the post-classification stream uses *wieghted sum* to aggregate snippet features. By such a linear operation, the aggregated feature remains in the original feature space. When an ideal classification mapping function is given, both pipelines would obtain accurate classification results. This inspires us to simultaneously learn these two pipelines under a newly proposed Equivalent Classification Mapping (ECM) mechanism to obtain the desired classifier and then use the classifier to identify the presence of the actions on each temporal point.

To implement such a new learning mechanism, the proposed method (see Fig. 1 (c)) has the following three main characteristics:

- Compiling the basic equivalent classification mapping spirit, we implement two parallel network streams to model the pre-classification pipeline and the post-classification pipeline, respectively, and make the two network streams share the same classifier.
- Besides the basic classification loss of each network stream, we introduce a new equivalent classification loss, which contains a classification-to-classification consistency term and an aggregation-to-classification consistency term to penalize the inconsistency of the classification scores from two network streams and the inconsistency of the classification score and the aggregation weights, respectively.
- Unlike the traditional post-classification pipelines [4, 5, 6] which infer the feature aggregation weights in self-attention-like manners, the post-classification stream in the proposed learning framework is equipped with a weight-transition module that transits the frame-level classification scores obtained from the pre-classification stream to generate the feature aggregation weights for the post-classification pipeline.

In summary, ECM contributes to the revealing of the equivalence mechanism, together with designing equivalence-based components. The equivalence mechanism indicates that both the classification for snippet features in the pre-classification pipeline and the classification for aggregated features in the post-classification pipeline pursue the same ideal classifier. Equivalence mechanism is overlooked by previous methods but plays an essential role, as verified in Section 4. Although conceptually simple, an intuitive implementation via sharing classifier only achieves 21.6. Thus, adequately mining the equivalence is critical for good results. Specifically, we design equivalence-based components, i.e., weight-transaction module and two consistency losses ($\mathcal{L}_{c2c}$ and $\mathcal{L}_{a2c}$). Empirically, ECM starts from two simple and widely used baselines (with performance 19.6 and 17.1), employs equivalence-based components, achieves 29.1 and builds new SOAT without bells and whistles. Moreover, ECM can serve as a clean and solid baseline for future studies on weakly supervised action localization.

2 Related work

Supervised action localization Early explorations adopt a detection-by-classification pipeline. S-CNN [7] classifies sliding-window proposals to localize action instances. CDC [8] and Lin *et al.* [9, 10] predict the actionness score for each temporal point. Meanwhile advances in object detection bring inspiration to action localization. Some methods follow Faster R-CNN [11] and perform two-stage action localization, e.g., R-C3D [12], CBR [13], TURN TAP [14] and TAL [15]. Similarly, some methods follow one-stage object detection methods [16, 17] and perform action localization, e.g., SSAD [18] and GTAN [19]. In addition, the recurrent memory module is used to capture long-term dependencies, such as SS-TAD [20], SST [21] and the work from Yuan *et al.* [22]. Apart from the above explorations, some noticeable works also include modeling temporal structure [23, 24], modeling context [15, 25, 26], modeling relationships among action proposals [27], localizing action instances from a part of videos [28], etc. In summary, the supervised methods can explicitly learn from segment-level annotations and achieve accurate localization. However, they are limited by the expensive annotations as well, which can be alleviated by the studied weakly supervised method.

Weakly supervised action localization UntrimmedNets [1] proposes a pre-classification pipeline with a soft aggregation mechanism. Later, W-TALC [29] considers the co-activity similarity to model inter-video similarities and differences. Furthermore, CMCS [2] proposes to jointly learn multiple classification networks and require them to generate diverse responses. Focused on the quality of the class activation sequence, there are some other promising works. CleanNet [30] learns regression to adjust the action segments. TSM [31] models the action structure via a multi-phase process. 3C-Net [3] introduces action count cues. BaSNet [32] tries to suppress the background response. In general, the foundation of pre-classification methods is the class activation sequence, generated by the classification mapping function.

Apart from the above methods, there is another pipeline, namely the post-classification pipeline. Inspired by the success of CAM [33], STPN [4] proposes to first aggregate video features then perform classification. A similar intuition can be found in ST-GradCAM [34]. Later, Nguyen *et al.* [5] extend STPN [4] by introducing background modeling. Meanwhile, MAAN [6] develops the post-classification pipeline with a marginalized average aggregation module. In general, the post-classification pipeline is aware of the complete video features, but its response to each category may be not discriminative enough.

Video recognition and spatio-temporal action detection Video recognition researches can provide high-level feature representations and bring inspiration for the action localization task. The widely used video recognition models include C3D [35], two-stream network [36], I3D [37], etc. Recently proposed video recognition models also include TrajectoryNet [38], SlowFast [39] and the work of Fan *et al.* [40]. Spatio-temporal action detection aims to detect action with the bounding box at each frame. Recent progress includes learning to track [41], VideoCapsuleNet [42], SlowFast networks [39], etc. In addition, some works consider more practical scenarios. For example, Choi *et al.* [43] propose to mitigate the sense bias when recognizing video. Chéron *et al.* [44] perform spatio-temporal action detection with varying types of weak supervision.

3 Method

Given a dataset containing C action categories and each video in the dataset has the classification label $\mathbf{y} = [y_1, y_2, ..., y_C]$, where $y_c \in \{0, 1\}, c \in [1, C]$ indicates whether there is an action instance belonging to category c. Each video is firstly divided into T snippets and then a feature extractor is used to extract the feature representation $\mathbf{f}_t \in \mathbb{R}^D, t \in [1, T]$ from each video snippet. Under the weakly supervised learning scenario, the proposed model learns the classifier $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_C]$ from the video-level classification label $\mathbf{y}$ to discover the temporal locations of the desired action instances from each video. When the training process is complete, we forward each input video through the learned classifier and obtain the class activation sequences. Then, following [32], we adopt a post-process to reject the categories whose classification scores are smaller than a threshold τ. Next, the remaining class activation sequences are min-max normalized along the temporal dimension, which is followed by the temporal actionness grouping operation [23] to localize action instances. Finally, redundant action instances are filtered out via Non-Maximum Suppression.

3.1 The body network

As shown in Fig. 1 (c), the body network consists of two streams: the pre-classification stream and the post-classification stream. They share the same classification mapping function but make predictions in different manners. To endow the classifier $\mathbf{\Phi}$ with the helpful temporal reception field, we use three temporal convolutional layers to build it (see Fig. 2), where the kernel sizes of the first two layers are 3 while it of the last layer is 1. Here, the classifier is used to predict the classification scores for each temporal point, i.e., each video snippet. Specifically, in the pre-classification stream, the extracted video features are directly passed through the classifier. Then, obtain classification scores along the temporal points and generate the class activation sequence for the whole video. After that, we adopt the top-k mean strategy to aggregate the classification scores for each category c and obtain the video-level classification score $\mathbf{s}^e = [s_1^e, s_2^e, ..., s_C^e]$.

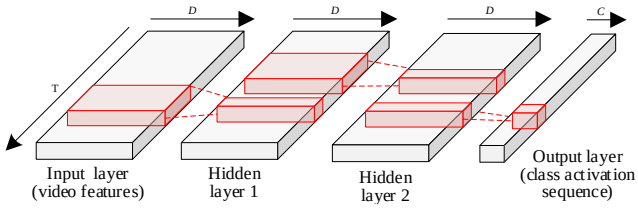

Figure 2: Architecture of the classifier used in our learning framework.

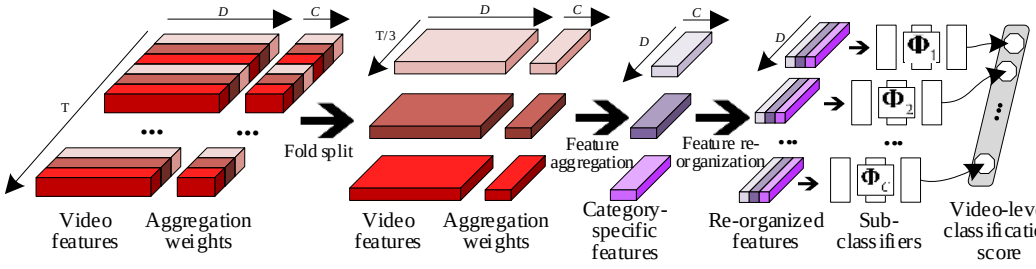

Figure 3: Illustration of the detailed operations in the post-classification stream in the proposed learning framework.

In the post-classification stream, the inputs consist of both the video features $\mathbf{F}$ and the category-specific aggregation weights. Here, we explore both the action and background weights $\mathbf{W}^a \in \mathbb{R}^{C \times T}, \mathbf{W}^b \in \mathbb{R}^{C \times T}$ to generate the category-specific action features $\mathbf{F}^a = [\mathbf{f}_1^a, \mathbf{f}_2^a, ..., \mathbf{f}_C^a]$ and category-specific background features $\mathbf{F}^b = [\mathbf{f}_1^b, \mathbf{f}_2^b, ..., \mathbf{f}_C^b]$, where $\mathbf{f}_c^* \in \mathbb{R}^{D \times 3}$. Specifically, the category-specific feature for either the action or the background is obtained by operating the weighted sum over the video features $\mathbf{F}$:

$$\mathbf{f}_c^a = \sum_{t=1}^{T} w_{c,t}^a \times \mathbf{f}_t, \quad \mathbf{f}_c^b = \sum_{t=1}^{T} w_{c,t}^b \times \mathbf{f}_t \tag{1}$$

where $w_{c,t}^a$ and $w_{c,t}^b$ are the elements in $\mathbf{W}^a$ and $\mathbf{W}^b$, respectively. Then, each sub-classifier $\mathbf{\Phi}_c$ is applied on the corresponding category-specific action feature $\mathbf{f}_c^a$ to obtain the final classification score $\mathbf{s}^o = [s_1^o, s_2^o, ..., s_C^o]$, where $s_c^o = f(\mathbf{f}_c^a | \mathbf{\Phi}_c)$, $f(\cdot)$ denotes the network forward operation.

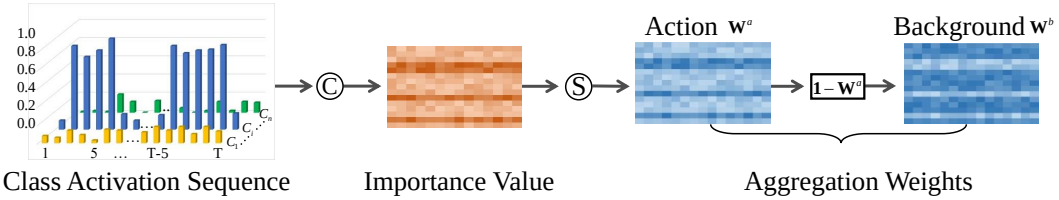

Figure 4: Equivalent weight-transition module. We predict the foreground and background aggregation weights from the class activation sequence. ⓒ indicates temporal convolution, Ⓢ indicates sigmoid activation.

Since the classifier uses the same structure as the one used in the pre-classification stream, we evenly divide the raw video features as well as the corresponding aggregation weights into three folds by sampling every three snippets, and implement the feature aggregation process within each fold (see Fig. 3). Comparing to the strategy that directly extracting the whole video features in one fold, the aggregated video-level features extracted by our strategy can better fit the input structure of the classifier, thus facilitating a more effective equivalent learning scheme.

3.2 Equivalent weight-transition module

The post-classification stream requires category-specific aggregation weights $\mathbf{W}^a$ and $\mathbf{W}^b$ to aggregate features. Considering that the class activation sequence obtained from the pre-classification stream can reveal the probability that a temporal point belongs to an action instance, we propose to obtain the aggregation weights by transiting the class activation sequence, as shown in Fig. 4. We use a convolutional layer with kernel size 1 to predict the importance values for each temporal point. After that, we apply the sigmoid activation and obtain the action aggregation weights $\mathbf{W}^a$, while the background aggregation weights can be obtained via $\mathbf{W}^b = \mathbf{1} - \mathbf{W}^a$. Compared with the conventional methods [4, 5, 6, 32], the proposed equivalent weight-transition module shows two characteristics. Firstly, existing works adopt an extra network to predict the aggregation weights, while the proposed module can directly obtain aggregation weights from the pre-classification stream. Besides, existing methods predict category-agnostic weights and use them to obtain one aggregated feature to represent the input video. In contrast, we learn category-specific weights and use them to obtain the aggregated features for each different action categories. Such category-specific features would bring richer representation and assist ECM to clearly distinguish different action categories, as verified in Section 4.3.

3.3 Equivalent classification loss

The equivalent classification loss $\mathcal{L}_{\text{equ}}$ consists of a classification-to-classification consistency term $\mathcal{L}_{\text{c2c}}$ and an aggregation-to-classification consistency term $\mathcal{L}_{\text{a2c}}$, i.e., $\mathcal{L}_{\text{equ}} = \mathcal{L}_{\text{c2c}} + \alpha\mathcal{L}_{\text{a2c}}$, where α is the coefficient. Given an input video, the pre-classification stream predicts the classification score $\mathbf{s}^e$, while the post-classification stream predicts the classification score $\mathbf{s}^o$ with action aggregation weights $\mathbf{W}^a$. Based on the intuition that the classification scores predicted by the two network streams should be identical for the same input video, we introduce the classification-to-classification consistency term $\mathcal{L}_{\text{c2c}}$ to the learning loss function:

$$\mathcal{L}_{\text{c2c}} = \frac{1}{C}\sum_{i=1}^{C}(s_i^{\text{e}} - s_i^{\text{o}})^2 \tag{2}$$

The classification-to-classification consistency term $\mathcal{L}_{\text{c2c}}$ plays a role in making one stream perceive the predictions from the other stream. Despite the simplicity, it obviously facilitates the learning of the classification mapping function, as shown in Section 4.3.

Besides classification-to-classification consistency, we further explore a new training strategy, namely aggregation-to-classification consistency training. The motivation is that the video-level classification score should be consistent with the aggregation weights. Specifically, to meet the conventional classification loss, the attention weights may only highlight the most discriminative snippets within an action. In contrast, the proposed aggregation-to-classification loss $\mathcal{L}_{\text{a2c}}$ not only requires the action attention weights $\mathbf{W}^a$ to highlight snippets within an action, but also requires the background attentions $\mathbf{W}^b = \mathbf{1} - \mathbf{W}^a$ to exclude all action snippets. This is not explored by classification loss.

Consider a video with classification label $\mathbf{y}$, the post-classification stream can predict the action presence score $\mathbf{s}^o = [s_1^o, s_2^o, ..., s_C^o]$ with action weights $\mathbf{W}^a$. Meanwhile it can predict the action absence score $\tilde{\mathbf{s}}^o = [\tilde{s}_1^o, \tilde{s}_2^o, ..., \tilde{s}_C^o]$ with background weights $\mathbf{W}^b$. Similar to the action presence score $\mathbf{s}^o$, the action absence score $\tilde{\mathbf{s}}^o$ is obtained by $\tilde{\mathbf{s}}_c^o = f(\mathbf{f}_c^b|\mathbf{\Phi}_c)$. Then, for an input video that contains k action categories, we select classification scores for present action categories from $\mathbf{s}^o$ and $\tilde{\mathbf{s}}^o$ according to classification label $\mathbf{y}$. After that, we define $\mathbf{s}' = [s_{k_1}^o, s_{k_2}^o, ..., s_{k_k}^o, \tilde{s}_{k_1}^o, \tilde{s}_{k_2}^o, ..., \tilde{s}_{k_k}^o]$, which contains k positive labels and k negative labels. Finally, we perform sigmoid activation over prediction $\mathbf{s}'$ and define the aggregation-to-classification consistency term $\mathcal{L}_{\mathrm{a2c}}$ as:

$$\mathcal{L}_{\mathrm{a2c}} = -\frac{1}{2k}[\sum_{i=1}^{k} \log(s_{k_i}^o) + \sum_{i=1}^{k} \log(1 - \tilde{s}_{k_i}^o))] \tag{3}$$

The proposed aggregation-to-classification consistency training strategy can guide ECM to better distinguish action segments and backgrounds. Besides, the existing background modeling strategy [5] and background suppression strategy [32] can be regarded as the specific cases of this strategy, as they only consider the background category when adjusting attention weights.

Finally, we also calculate the basic classification loss $\mathcal{L}_{\mathrm{cls}} = \mathcal{L}_{\mathrm{cls,e}} + \mathcal{L}_{\mathrm{cls,o}}$, where $\mathcal{L}_{\mathrm{cls,e}}$ and $\mathcal{L}_{\mathrm{cls,o}}$ are the classification loss from the pre-classification stream and the post-classification stream, respectively. Formally, this problem can be formulated as a multi-label classification problem. We perform $L1-$normalization on the original classification label $\mathbf{y}$ and calculate the cross-entropy loss $\mathcal{L}_{\mathrm{cls,e}}$ and $\mathcal{L}_{\mathrm{cls,o}}$. In summary, the overall loss function required to be optimized is $\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \beta\mathcal{L}_{\mathrm{equ}}$, where β is the balance coefficient.

4 Experiments

4.1 Experimental setups

Dataset We perform experiments on three benchmarks: THUMOS14 [45], ActivityNet v1.2 [46] and ActivityNet v1.3 [46]. THUMOS14 consists of 20 action categories, including 200 videos for training and 213 videos for testing. ActivityNet v1.2 consists of 100 action categories, 9682 videos. The video number ratio among training, validation and testing sets is 2:1:1. ActivityNet v1.3 is an extension of ActivityNet v1.2, with 200 categories and 19994 videos.

Metric The evaluation is performed under the official metric of each dataset, i.e., mean Average Precision (mAP). THUMOS14 focuses on mAP under threshold 0.5, while ActivityNet focuses on the average mAP under thresholds [0.50:0.05:0.95]. Following previous works [4, 5, 32], we adopt the official evaluation tools of the ActivityNet dataset to perform evaluations.

Feature extraction To extract features, each video is evenly divided into T snippets and we uniformly sample 16 frames from each snippet, similar as previous works [4, 5]. The optical flow is calculated via the TV-L1 [47] algorithm. I3D model [37] pre-trained on the Kinetics-400 dataset is used to extract video features, without finetuning on THUMOS14 [45] or ActivityNet v1.3 [46]. We extract both the appearance features and the motion features, which are concatenated together to represent the video sequence. The concatenated feature dimension is 2048. Finally, the snippet number for THUMOS14, ActivityNet v1.2 and ActivityNet v1.3 are 750, 100 and 100, respectively.

Training and evaluation details ECM is implemented using PyTorch [48]. Adam [49] solver is used to optimize the network. For all experiments, we set batch size as 16, the learning rate as 2×10^{-4} and the weight decay as 5×10^{-4}. We train ECM for 150 epochs, 40 epochs and 40 epochs for THUMOS14, ActivityNet v1.2 and ActivityNet v1.3, respectively. The parameters are empirically determined via gird search. Specifically, we set $k = \frac{1}{8}$ for the top-k mean aggregation. The balance coefficients are $\alpha = 100$ and $\beta = 0.05$. In evaluation, the threshold to reject absent categories is $\tau = 0.25$. The implementation code is in the supplementary materials.

Table 1: Comparisons between ECM and the state-of-the-art methods on THUMOS14 dataset. The performance for both fully supervised methods and weakly supervised methods are reported. Most methods adopt the I3D feature, while there are also two-stream feature (TS) and UntrimmedNet feature (UNT). We report mAP under different thresholds, as well as the average mAP under threshold [0.1:0.7] and [0.3:0.7].

Sup.	Method	Fea.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.1:0.7	0.3:0.7
Full	CDC [8]	-	-	-	40.1	29.4	23.3	13.1	7.9	-	22.8
	R-C3D [12]	-	54.5	51.5	44.8	35.6	28.9	-	-	-	-
	BMN [10]	TS	-	-	56.0	47.4	38.8	29.7	20.5	-	38.5
	TAL [15]	I3D	59.8	57.1	53.2	48.5	42.8	**33.8**	**20.8**	**45.1**	**39.8**
	PGCN [27]	I3D	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**	-	-	-	-
Weak	STPN [4]	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	27.0	18.5
	MAAN [6]	I3D	59.8	50.8	41.1	30.6	20.3	12.0	6.9	31.6	22.2
	AutoLoc [50]	UNT	-	-	35.8	29.0	21.2	13.4	5.8	-	21.0
	CMCS [2]	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.4	23.7
	CleanNet [30]	I3D	-	-	37.0	30.9	23.9	13.9	7.1	-	22.6
	TSM [31]	I3D	-	-	39.5	31.9	24.5	13.8	7.1	-	23.4
	3C-Net [3]	I3D	59.1	53.5	44.2	34.1	26.6	-	8.1	-	-
	WLBM [5]	I3D	60.4	**56.0**	**46.6**	37.5	26.8	17.6	9.0	36.3	27.5
	BaSNet [32]	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.3	27.3
	ECM	I3D	**62.6**	55.1	46.5	**38.2**	**29.1**	**19.5**	**10.9**	**37.4**	**28.8**

Table 2: Comparisons on ActivityNet v1.2 dataset.

Sup.	Method	Fea.	0.50	0.75	0.95	avg
Full	SSN [23]	TS	**41.3**	**27.0**	**6.1**	**26.6**
Weak	TSM [31]	I3D	28.3	17.0	3.5	17.1
	W-TALC [29]	I3D	37.0	-	-	18.0
	3C-Net [3]	I3D	37.2	-	-	21.7
	CleanNet [30]	I3D	37.1	20.3	5.0	21.6
	CMCS [2]	I3D	36.8	22.0	5.6	22.4
	BaSNet [32]	I3D	38.5	24.2	5.6	24.3
	ECM	I3D	**41.0**	**24.9**	**6.5**	**25.5**

Table 3: Comparisons on ActivityNet v1.3 dataset.

Sup.	Method	Fea.	0.5	0.75	0.95	avg
Full	GTAN [19]	P3D	**52.6**	**34.1**	**8.9**	**34.3**
Weak	STPN [4]	I3D	29.3	16.9	2.6	-
	TSM [2]	I3D	30.0	19.0	4.5	-
	MAAN [6]	I3D	33.7	21.9	5.5	-
	CMCS [31]	I3D	34.0	20.9	5.7	21.2
	BaSNet [5]	I3D	34.5	22.5	4.9	22.2
	WLBM [32]	I3D	36.4	19.2	2.9	-
	ECM	I3D	**36.7**	**23.6**	**5.9**	**23.5**

4.2 Comparison with state-of-the-arts

Experiments on THUMOS14 In Tab. 1, we compare ECM with recent state-of-the-art temporal action localization methods, including both the weakly supervised methods and the supervised ones. Evaluated by the official metric, mAP under threshold 0.5, ECM exceeds the strong competitor BaSNet [32] by a margin of 2.1. Besides, WLBM [5] achieves high performance under threshold 0.2 and 0.3. For a comprehensive comparison, we follow the metric used by the ActivityNet dataset and calculate average mAP under thresholds [0.1:0.7]. This metric demonstrates that ECM is superior to both WLBM [5] and BaSNet [32]. Similarly, some methods ([30, 31, 50]) only report scores under thresholds [0.3:0.7]. We calculate the average mAP under the same thresholds and find ECM exceeds the previous state-of-the-art method WLBM [5] by a margin of 1.3. Furthermore, we make comparisons between the weakly supervised methods and the fully supervised methods, considering mAP under threshold 0.5. On one hand, ECM improves the performance of the weakly supervised method and exceeds some supervised methods [8, 12]. On the other hand, the performance gap is 20.0 when compared with state-of-the-art supervised method PGCN [27], indicating the weakly supervised methods should be continuously studied in the future.

Experiments on ActivityNet v1.2 Tab. 2 reports the performance on ActivityNet v1.2. Under the metric average mAP, BaSNet [32] shows strong performance 24.3, which is exceeded by ECM with

Table 4: Ablation studies about network architecture and loss functions.

Pre-Cls	✓		✓	✓	☑	✓
Post-Cls		✓	✓	✓	☑	✓
$\mathcal{L}_{a2c}$				✓	✓	✓
$\mathcal{L}_{c2c}$					✓	✓
mAP	19.6	17.1	22.8	24.2	23.4	29.1

Table 5: Ablation studies about classifier and attention.

Classifier		Attention		mAP
Individual	Sharing	Learning	Transiting	
✓		✓		21.4
	✓	✓		21.6
	✓		✓	22.8

a margin of 1.2. Besides, under threshold [0.50, 0.75, 0.95], ECM consistently improves the previous state-of-the-art performance.

Experiments on ActivityNet v1.3 As shown in Tab. 3, ECM achieves superior performance on both the average mAP and mAP under threshold [0.50, 0.75, 0.95], when compared with weakly supervised methods. Furthermore, although WLBM [5], BaSNet [32] and CMCS [2] show good performance under threshold [0.50, 0.75, 0.95] respectively, they are all exceeded by ECM.

4.3 Ablation studies

We perform ablation studies on the THUMOS14 dataset and the results are reported in Tab. 4. Under threshold 0.5, the pre-classification stream and post-classification stream achieve 19.6 and 17.1, respectively. Moreover, we share the classifier between the pre-classification stream and the post-classification stream, transit the aggregation weights from the class activation sequence in the pre-classification stream, and the performance reaches 22.8. Based on this, the proposed aggregation-to-classification consistency training strategy can lift the performance to 24.2, demonstrating it is rational to constrain the consistency between the aggregation weights and the action presence. Finally, the complete ECM achieves 29.1, under the cooperation of sharing the classifier, the equivalent classification loss and the weight-transition module. To verify the effectiveness of classifier sharing, we remove classifier sharing from ECM and keep all other components unchanged. Under this setting, the model obtains 23.4, which is much inferior to 29.1 of full ECM model. This demonstrates classifier sharing (i.e., the equivalence mechanism) is the foundation of ECM, which makes the proposed equivalence-based components work effectively.

In addition, we study different strategies to integrate the pre-classification stream and the post-classification stream together. A simple strategy is to directly fuse the localization results from these two streams. We elaborately try different fusing methods: (1) First localizing action instances from two class activation sequences (CASs), then merging the localization results. (2) First fusing two CASs via weighted sum (we try different weights), then localizing action instances from the fused CAS[1]. Actually, the above experiments are ensemble methods, where we train two individual classifiers and ensemble the localization results. The best performance is 21.4. Based on the equivalence mechanism, we share the classifier between the pre-classification stream and the post-classification stream, the performance is 21.6. Moreover, if the attention weight is transited from the pre-classification stream, the performance would be 22.8. Above experiments demonstrate the rationality of sharing classifier and weight transition.

Beyond that, we verify the effectiveness of class-specific feature representation. We follow existing methods [4, 5, 6] and learn class-agnostic aggregation weights for the post-classification stream, such a variant gets 27.6. In comparison, the 1.5 performance improvements of ECM is mainly caused by rich feature representation.

To make a thorough analysis about ECM, we use the diagnosing tools [51] analyze the localization results. The results are shown in Tab. 6. ECM improves true positives and eliminates background errors. Meaning of metrics: TP: True Positive, BG: Background Error, CON: Confusion Error, DD: Double Detection Error, LOC: Localization Error, WL: Wrong Label Error.

Table 6: Diagnosing localization results of ECM, measured by mAP@0.5 on THUMOS14 dataset.

	TP↑	BG↓	CON↓	DD↓	LOC↓	WL↓
ECM	**14.66**	**61.38**	**3.38**	1.81	17.86	**0.91**
Pre	9.16	66.53	4.09	1.65	**17.51**	1.06
Post	9.33	65.76	4.89	**1.15**	17.85	1.03

[1]Details in supplementary materials.

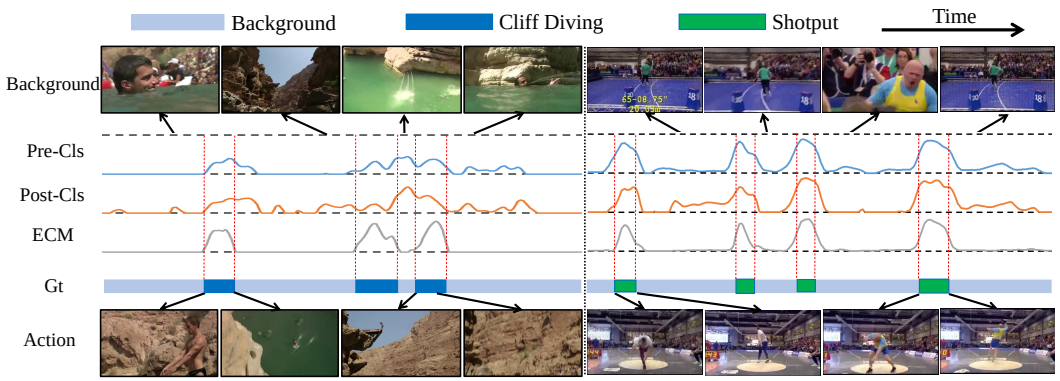

Figure 5: Qualitative results of the class activation sequences for pre-classification stream (Pre-Cls), post-classification stream (Post-Cls) and the proposed ECM.

We qualitatively visualize the class activation sequences in Fig. 5. When only using the pre-classification stream or the post-classification stream, the generated class activation sequences cannot distinguish two adjacent action instances (see the left case), or shows high responses for a part of background points (see the right case). This will lead to error action localizations. In contrast, the proposed ECM method precisely shows high responses for the action segments and confidently suppresses the response of backgrounds, leading to accurate localization results.

5 Conclusion

In this paper, we introduce the equivalent classification mapping mechanism to weakly supervised temporal action localization task, which starts from both the pre-classification stream and the post-classification stream to simultaneously learn one classification mapping function. Assisted with the classification equivalent loss and the weight-transition module, ECM achieves state-of-the-art performance on three benchmarks. Considering ECM is simple to implement and achieves good performance, subsequent researches are potential to observe further improvements when applying the ECM mechanism to the pre-classification methods [2, 3, 32] and post-classification methods [4, 5, 6]. Besides, we plan to verify the performance of ECM on instructional videos [52] or large scale action localization datasets [53, 54]. Furthermore, it is a promising direction that extending ECM to other similar research areas, for example, weakly supervised object localization [33] and detection [55].

Broader Impact

We present a novel method to discover actions from untrimmed videos. As three benchmark datasets show large varieties, the proposed method does not leverage biases in the data, which is further verified by the performance improvements on all three datasets. Because it only requires video-level labels to train, the proposed method is potential to be applied to internet video platforms, e.g., YouTube, to tackle the large-scale, fast-growing untrimmed videos. Besides, the proposed method can help to discover informative video segments, and filter out most of the meaningless backgrounds. For example, the proposed method can help to build a smart video surveillance system, which can be used for monitoring the patients, monitoring the operation of the production line, etc.

On the contrary, it should also be noticed that employees working on video monitoring may be put at disadvantage from this research. The automatic video processing and understanding system would reduce existing jobs, while it may create new jobs that can alleviate this scenario. Because the proposed method directly learns from the video-level category label, the failure cases mainly come from incomplete or missing localizations. Currently, the localization quality is not comparable to fully supervised methods, but it can be gradually improved when more effective algorithms are proposed, and large-scale videos and powerful computation ability is available to train the model.

References

[1] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334, 2017.

[2] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, pages 1298–1307, 2019.

[3] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *ICCV*, pages 8679–8687, 2019.

[4] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pages 6752–6761, 2018.

[5] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, pages 5502–5511, 2019.

[6] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *ICLR*, 2019.

[7] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016.

[8] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 5734–5743, 2017.

[9] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–19, 2018.

[10] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[12] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5783–5792, 2017.

[13] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *BMVC*, 2017.

[14] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, pages 3628–3636, 2017.

[15] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139, 2018.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[18] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *ACM MM*, pages 988–996, 2017.

[19] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, pages 344–353, 2019.

[20] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, volume 2, page 7, 2017.

[21] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, pages 2911–2920, 2017.

[22] Jun Yuan, Bingbing Ni, Xiaokang Yang, and Ashraf A Kassim. Temporal action localization with pyramid of score distribution features. In *CVPR*, pages 3093–3102, 2016.

[23] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017.

[24] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *CVPR*, pages 3684–3692, 2017.

[25] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *ICCV*, pages 5793–5802, 2017.

[26] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *ECCV*, pages 68–83, 2018.

[27] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, pages 7094–7103, 2019.

[28] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. Action search: Spotting actions in videos and its application to temporal action localization. In *ECCV*, pages 251–266, 2018.

[29] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 563–579, 2018.

[30] Ziyi Liu, Le Wang, Qilin Zhang, Zhanning Gao, Zhenxing Niu, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*, pages 3899–3908, 2019.

[31] Tan Yu, Zhou Ren, Yuncheng Li, Enxu Yan, Ning Xu, and Junsong Yuan. Temporal structure mining for weakly supervised action detection. In *ICCV*, pages 5522–5531, 2019.

[32] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, 2020.

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[34] Yunlu Xu, Chengwei Zhang, Zhanzhan Cheng, Jianwen Xie, Yi Niu, Shiliang Pu, and Fei Wu. Segregated temporal assembly recurrent networks for weakly supervised multiple action detection. In *AAAI*, volume 33, pages 9070–9078, 2019.

[35] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV Workshops*, pages 3154–3160, 2017.

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, pages 568–576, 2014.

[37] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[38] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *NeurIPS*, pages 2204–2215, 2018.

[39] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[40] Quanfu Fan, Chun-Fu Richard Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. In *NeurIPS*, pages 2261–2270, 2019.

[41] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015.

[42] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *NeurIPS*, pages 7610–7619, 2018.

[43] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *NeurIPS*, pages 851–863, 2019.

[44] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A flexible model for training action localization with varying levels of supervision. In *NeurIPS*, pages 942–953, 2018.

[45] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014.

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

[47] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[50] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, pages 154–171, 2018.

[51] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, pages 256–272, 2018.

[52] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, pages 1207–1216, 2019.

[53] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*, pages 8668–8678, 2019.

[54] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[55] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 128(2):261–318, 2020.

[56] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, pages 6508–6516, 2018.

[57] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*, pages 3546–3555, 2019.

[58] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[60] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[61] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017.

[62] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[63] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.

[64] Xiao-Yu Zhang, Haichao Shi, Changsheng Li, Kai Zheng, Xiaobin Zhu, and Lixin Duan. Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision. In *AAAI*, volume 33, pages 9227–9234, 2019.

[65] Chengwei Zhang, Yunlu Xu, Zhanzhan Cheng, Yi Niu, Shiliang Pu, Fei Wu, and Futai Zou. Adversarial seeded sequence growing for weakly-supervised temporal action localization. In *ACM MM*, pages 738–746, 2019.

[66] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, pages 3604–3613, 2019.

[67] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, pages 3544–3553. IEEE, 2017.

Supplymentary Material

In this supplementary material, Section "*Implementation of the post-classification stream*" provides details about the implementation of the post-classification stream. Section "*Details of the ablation studies*" elaborates on the setups and results for the ablation studies. Next, we report frame accuracy in Section "*Frame accuracy*", while the detailed comparisons on three benchmarks are shown in Section "*Detailed comparisons*". Finally, more qualitative results are shown in Section "*Qualitative results*".

Implementation of the post-classification stream

The post-classification stream shares the same classifier as the pre-classification stream, which consists of three convolutional layers. We show the implementation details of each operation as well as the size for the corresponding output tensor in Tab. 7. Specifically, each category-specific feature contains three feature vectors. The first two temporal convolutional layers adopt the ordinary temporal convolution to learn feature representation from the category-specific features. Then, the last layer adopts the group temporal convolution to predict the category-specific classification score. Finally, we perform average pooling along the temporal dimension to generate the video-level classification score.

Table 7: Implementation details of the post-classification stream. B: batch size; C: category number; D: feature dimension.

Operation	Output Size
–	B×C×D×3
Reshape	(B×C)×D×3
Temporal Convolution	(B×C)×D×3
Temporal Convolution	(B×C)×D×3
Reshape	B×(C×D)×3
Group Temporal Convolution	B×C×3
Average Pooling	B×C

Details of the ablation studies

Performance of the post-classification stream

To validate the performance of the post-classification stream, we need an extra network to predict the category-specific aggregation weights. Based on previous works [4, 32], we try different network architectures to determine a proper one. In Tab. 8, the network architecture is represented in the form of "(input)-middle layer-[output]". For example, "(2048)-512-[20]" indicates that the input is a 2048-D feature, the middle layer is a convolutional layer with 512 filters, and the output is a 20-D classification score.

We find that a proper architecture to infer the aggregation weights might be (2048)-2048-2048-[20], where there are two middle layers with 2048 filters. This setting achieves mAP=17.1 under threshold 0.5 and average mAP=18.6 under threshold [0.1:0.7]. When we use more parameters, e.g., using convolutional layers with kernel size 3, the performance drops. Thus, we report 17.1 for the performance of the post-classification stream.

Fuse localization results from two separate streams

In ablation studies, when the pre-classification stream and the post-classification stream do not share the classifier (indicated as "☑" in Tab. 4 of the paper), the evaluation process would obtain two different class activation sequences. Then, we study two ways to fuse the class activation sequences, where the first one is "concatenating localization results" (Concat) while the second one is "weighted sum class activation sequence" (WeightedSum).

Given a video, we feed it through both the pre-classification stream and the post-classification stream, obtain two class activation sequences. As for Concat, we individually obtain two groups of action localization results via thresholding on two class activation sequences. Then we concatenate the two results together and apply Non-Maximum Suppression to generate the final localization results. As for WeightedSum, this strategy is proposed by previous works [4, 5]. It fuses two class activation sequences via weighted sum and then localizes actions based on the obtained class action sequence.

Table 8: Performance of the post-classification stream with different network architecture for inferring the aggregation weights. The results are measured by mAP and obtained from the THUMOS14 dataset.

Kernel Size	Network	0.1	0.2	0.3	0.4	0.5	0.6	0.7	avg
1	(2048)-512-[20]	40.3	33.1	26.4	21.1	15.9	9.8	4.7	17.0
	(2048)-1024-[20]	41.2	34.5	28.1	22.0	16.2	10.0	5.1	17.7
	(2048)-2048-[20]	41.5	34.5	26.6	21.1	16.3	10.4	5.7	17.6
	(2048)-1024-512-[20]	43.8	37.0	29.5	22.6	16.2	9.7	5.1	18.4
	(2048)-1024-1024-[20]	43.3	35.4	27.9	21.4	15.7	9.4	5.2	17.8
	(2048)-2048-1024-[20]	42.5	34.6	26.8	20.8	15.2	9.9	4.8	17.4
	(2048)-2048-2048-[20]	42.3	35.8	29.1	23.1	**17.1**	11.0	6.2	**18.6**
3	(2048)-2048-2048-[20]	42.4	35.1	28.1	21.1	15.0	9.3	5.0	17.5

The weights for the pre-classification stream and the post-classification stream are λ and $1-\lambda$. We report the experimental results with λ varies in $[0.2, 0.4, 0.6, 0.8]$.

The experimental results are reported in Tab. 9. It can be found that all fusion methods exceed the individual pre-classification stream or post-classification stream. WeightedSum achieves 20.8 and Concat achieves 21.4. We report the performance 21.4 in our paper.

Table 9: Fusion localization results from the pre-classification stream and the post-classification stream.

TH	Pre-Cls	Post-Cls	Concat	WeightedSum (λ)			
				0.2	0.4	0.6	0.8
0.1	46.1	42.3	45.5	43.5	44.2	44.9	45.1
0.2	38.8	35.8	39.2	36.6	37.4	38.2	38.4
0.3	32.1	29.1	33.0	30.7	31.2	31.9	32.1
0.4	25.3	23.1	26.5	24.9	25.5	26.0	26.3
0.5	**19.6**	**17.1**	**21.4**	20.2	20.3	20.5	20.8
0.6	12.7	11.0	13.8	13.4	13.9	14.3	14.4
0.7	7.2	6.2	7.9	8.2	8.2	8.3	8.4
avg	20.5	18.6	21.2	20.1	20.4	20.9	21.0

Frame accuracy

Table 10: Frame accuracy on THUMOS14 dataset.

	Pre-Classification	Post-Classification	ECM
Frame Accuracy	42.7%	39.0%	54.3%

All of the pre-classification method, the post-classification method and the proposed ECM method directly learn classifiers to predict the classification score for each temporal point. Following the *frame accuracy* metric adopted by the action segment researches [56, 57], we measure the frame accuracy of the generated class activation sequences, so as to validate the quality of the point-level classification. Similar to previous works [56, 57], we do not consider the background category in order to prevent the case that a method predicts all frames as background but still achieves high performance.

Given the obtained class activation sequences, we first predict the video-level classification scores and reject categories whose classification scores are lower than threshold $\tau = 0.25$. Then, for the remaining responses, we perform max pooling at the temporal dimension and obtain the point-wise classification predictions. Finally, the frame-wise classification accuracy is reported in Tab. 10. It

can be found that the pre-classification method performs somewhat better than the post-classification method, with a margin of 3.7%. The proposed ECM brings obvious improvement, with a margin of 11.6%. The results demonstrate that the proposed equivalent classification mapping mechanism can learn a high-quality classifier and can discover more accurate action snippets from backgrounds.

Detailed comparisons

Feature representations

As the feature representation plays an important role in most temporal action localization methods, we discuss the features used by each work for fair comparisons in Tab. 11, Tab. 12 and Tab. 13. "I3D" indicates the Inflated 3D features [37]. "TS" indicates the two-stream [36] features. Xiong et al. [58] design a two-stream network architecture for the video classification task, which is used to extract two-stream features by latter works. Specially, the spatial network uses the ResNet architecture [59] and the temporal network uses the BN-Inception architecture [60]. Both the I3D model and the two-stream model are trained on the Kinetics-400 dataset. "P3D" indicates pseudo-3D [61] features. "C3D" indicates 3D convolutional [62] features. "UNT" indicates UntrimmedNet [1] features, with the model architecture illustrated in TSN [63]. "Res" indicates that LTSR [64] uses ResNet [59] to extract the feature for each video frame. Apart from extracting features, there are some works directly learn from video frames, e.g, R-C3D [12], CDC [8] and S-CNN [7]. As most of the weakly supervised methods use the I3D feature, we implement our experiments with the I3D feature for fair comparisons.

Comparison experiments on THUMOS14

In Tab. 11, we elaborately compare ECM with recent temporal action localization methods, including both supervised methods and weakly-supervised methods. As the official evaluation metric of the THUMOS14 dataset is the mAP under threshold 0.5, we sort all the methods according to this. In Tab. 11, "Weak†" indicates the methods rely on extra annotations apart from video classification labels. Both STAR [34] and 3C-Net [3] rely on the action count information to localize action instances.

It can be found that ECM performs superior among the weakly supervised works under threshold 0.5. It is promising to notice that ECM can exceed some supervised works published in the early years, e.g., ICCV 2017 and CVPR 2017.

Among the weakly supervised works, we notice STAR [34] and ASSG [65] achieve good performance under low thresholds. STAR [34] integrates the temporal gradient-weighted CAM mechanism and the attention mechanism to localize action instances. Apart from the commonly used video-level classification labels, it also relies on the action frequency annotation to train the model. With the metric of average mAP under threshold [0.1:0.5], the reported performance is 44.0 without the use of action frequency. In contrast, under the same metric, ECM achieves 46.3. ASSG [65] first initializes seed regions from the class activation sequences, which are generated by a well-performed temporal action localization method. Then, it progressively extends the seed regions and learns the classification model. However, the performance of ASSG [65] heavily relies on the quality of initialization. Consequently, when initializing it by STPN [4], it only achieves 20.9 under threshold 0.5. Furthermore, the heavy dependence of initialization would limit its application to practical scenarios.

Comparison experiments on ActivityNet v1.2 and ActivityNet v1.3

Tab. 12 and Tab. 13 show comparisons between ECM and the existing temporal action localization methods on ActivityNet v1.2 and ActivityNet v1.3, respectively. We report the performance under each threshold as well as the average mAP under threshold [0.50:0.05:0.95].

Table 11: Comparison to the existing temporal action localization methods on the THUMOS14 dataset. The results are measured by mAP and sorted by the performance under threshold 0.5. "†" indicates using extra annotations apart from video classification labels.

Sup.	Method	Pub.	Fea.	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Full	TCN [25]	ICCV 17	TS	-	-	33.3	25.6	15.9	9.0	-
	S-CNN [7]	CVPR 16	-	47.7	43.5	36.3	28.7	19.0	10.3	5.3
	SST [21]	CVPR 17	C3D	-	-	37.8	-	23.0	-	-
	CDC [8]	CVPR 17	-	-	-	40.1	29.4	23.3	13.1	7.9
	SSAD [18]	ACM MM 17	C3D	50.1	47.8	43.0	35.0	24.6	-	-
	TURN [14]	ICCV 17	TS	54.0	50.9	44.1	34.9	25.6	-	-
	R-C3D [12]	ICCV 17	-	54.5	51.5	44.8	35.6	28.9	-	-
	SSN [23]	ICCV 17	TS	60.3	56.2	50.6	40.8	29.1	-	-
	SS-TAD [20]	BMVC 17	C3D	-	-	45.7	-	29.2	-	9.6
	CTAP [26]	ECCV 18	TS	-	-	-	-	29.9	-	-
	CBR [13]	BMVC 17	TS	60.1	56.7	50.1	41.3	31.0	19.1	9.9
	BSN [9]	ECCV 18	TS	-	-	53.5	45.0	36.9	28.4	20.0
	MGG [66]	CVPR 19	I3D	-	-	53.9	46.8	37.4	29.5	21.3
	BMN [10]	ICCV 19	TS	-	-	56.0	47.4	38.8	29.7	20.5
	GTAN [19]	CVPR 19	P3D	69.1	63.7	57.8	47.2	38.8	-	-
	TAL [15]	CVPR 18	I3D	59.8	57.1	53.2	48.5	42.8	33.8	20.8
	PGCN [27]	ICCV 19	I3D	69.5	67.8	63.6	57.8	**49.1**	-	-
Weak†	STAR [34]	AAAI 19	I3D	68.8	60.0	48.7	34.7	23.0	-	-
	3C-Net [3]	ICCV 19	I3D	59.1	53.5	44.2	34.1	26.6	-	8.1
Weak	Hide-and-Seek [67]	ICCV 17	C3D	36.4	27.8	19.5	12.7	6.8	-	-
	UntrimmedNetS [1]	CVPR 17	UNT	44.4	37.7	28.2	21.1	13.7	-	-
	STPN [4]	CVPR 18	I3D	52	44.7	35.5	25.8	16.9	9.9	4.3
	LTSR [64]	AAAI 19	ResNet-101	55.9	46.9	38.3	28.1	18.6	11.0	5.6
	MAAN [6]	ICLR 19	I3D	59.8	50.8	41.1	30.6	20.3	12.0	6.9
	AutoLoc [50]	ECCV 18	UNT	-	-	35.8	29.0	21.2	13.4	5.8
	W-TALC [29]	ECCV 18	I3D	55.2	49.6	40.1	31.1	22.8	-	-
	CMCS [2]	CVPR 19	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0
	CleanNet [30]	ICCV 19	I3D	-	-	37.0	30.9	23.9	13.9	7.1
	TSM [31]	ICCV 19	I3D	-	-	39.5	31.9	24.5	13.8	7.1
	ASSG [65]	ACM MM 19	I3D	65.6	59.4	50.4	38.7	25.4	15.0	6.6
	WLBM [5]	ICCV 19	I3D	60.4	56.0	46.6	37.5	26.8	17.6	9.0
	BaSNet [32]	AAAI 19	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4
	ECM		I3D	62.6	55.1	46.5	38.2	**29.1**	19.5	10.9

Table 12: Comparison to the existing temporal action localization methods on ActivityNet v1.2 dataset, measured by mAP and sorted by the average mAP under threshold $[0.50{:}0.05{:}0.95]$. "†" indicates using extra annotations apart from video classification labels.

Sup.	Method	Pub.	Fea.	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	avg
Full	SSN [23]	ICCV 17	TS	41.3	-	-	-	-	27.0	-	-	-	6.1	**26.6**
Weak†	3C-Net [3]	ICCV 19	I3D	35.4	-	-	-	-	22.9	-	-	-	8.5	21.1
Weak	AutoLoc [50]	ECCV 18	UNT	27.3	24.9	22.5	19.9	17.5	15.1	13.0	10.0	6.8	3.3	16.0
	TSM [31]	ICCV 19	I3D	28.3	26.0	23.6	21.2	18.9	17.0	14.0	11.1	7.5	3.5	17.1
	W-TALC [29]	ECCV 18	I3D	37.0	-	-	-	14.6	-	-	-	-	-	18.0
	CleanNet [30]	ICCV 19	I3D	37.1	33.4	29.9	26.7	23.4	20.3	17.2	13.9	9.2	5.0	21.6
	CMCS [2]	CVPR 19	I3D	36.8	-	-	-	-	22.0	-	-	-	5.6	22.4
	BaSNet [32]	AAAI 20	I3D	38.5	-	-	-	-	24.2	-	-	-	5.6	24.3
	ECM		I3D	41.0	37.7	34.2	31.5	28.5	24.9	21.2	17.0	12.1	6.5	**25.5**

Table 13: Comparison to the existing temporal action localization methods on ActivityNet v1.3 dataset, measured by mAP and sorted by the average mAP under threshold [0.50:0.05:0.95]. "†" indicates using extra annotations apart from video classification labels.

Sup.	Method	Pub.	Fea.	0.5	0.75	0.95	0.50:0.95
Full	R-C3D [12]	ICCV 17	-	26.8	-	-	12.7
	TCN [25]	ICCV 17	TS	36.4	21.2	3.9	-
	TAL [15]	CVPR 18	I3D	38.2	18.3	1.3	20.2
	CDC [8]	CVPR 17	-	45.3	26.0	0.2	23.8
	PGCN [27]	ICCV 19	I3D	48.3	33.2	3.3	31.1
	BSN [9]	ECCV 18	TS	52.5	33.5	8.9	33.7
	BMN [10]	ICCV 19	TS	50.1	34.8	8.3	33.9
	GTAN [19]	CVPR 19	P3D	52.6	34.1	8.9	**34.3**
Weak†	STAR [34]	AAAI 19	I3D	31.1	18.8	4.7	-
Weak	STPN [4]	CVPR 18	I3D	29.3	16.9	2.6	-
	TSM [31]	ICCV 19	I3D	30.0	19.0	4.5	-
	ASSG [65]	ACMMM 19	I3D	32.3	20.1	4.0	-
	LTSR [64]	AAAI 19	Res	33.1	18.7	3.3	21.8
	MAAN [6]	ICLR 19	I3D	33.7	21.9	5.5	-
	CMCS [2]	CVPR 19	I3D	34.0	20.9	5.7	21.2
	WLBM [5]	ICCV 19	I3D	36.4	19.2	2.9	-
	BaSNet [32]	AAAI 20	I3D	34.5	22.5	4.9	22.2
	ECM		I3D	36.7	23.6	5.9	**23.5**

Qualitative results

We show qualitative results of the class activation sequences and the corresponding ground truth for THUMOS14, ActivityNet v1.2 and ActivityNet v1.3 in Fig. 6, Fig. 7 and Fig. 8, respectively.

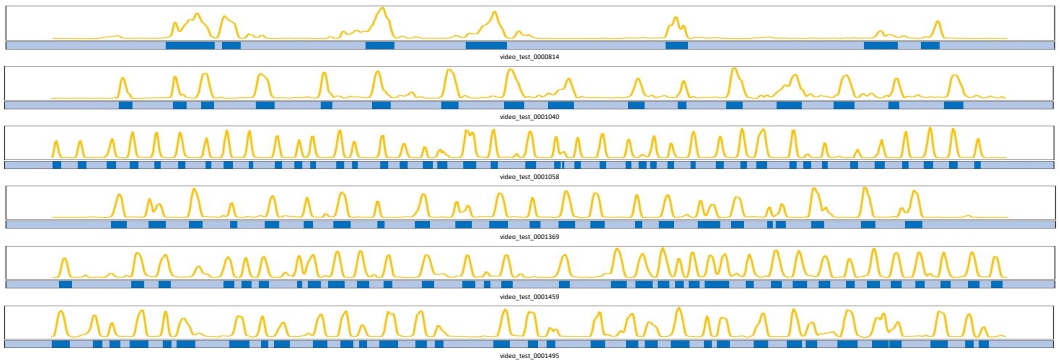

Figure 6: Qualitative results of the class activation sequences for ECM, on the THUMOS14 dataset.

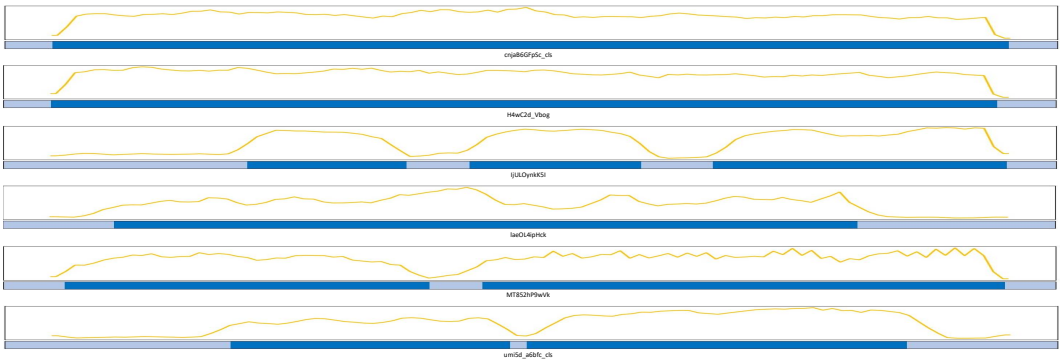

Figure 7: Qualitative results of the class activation sequences for ECM, on the ActivityNet v1.2 dataset.

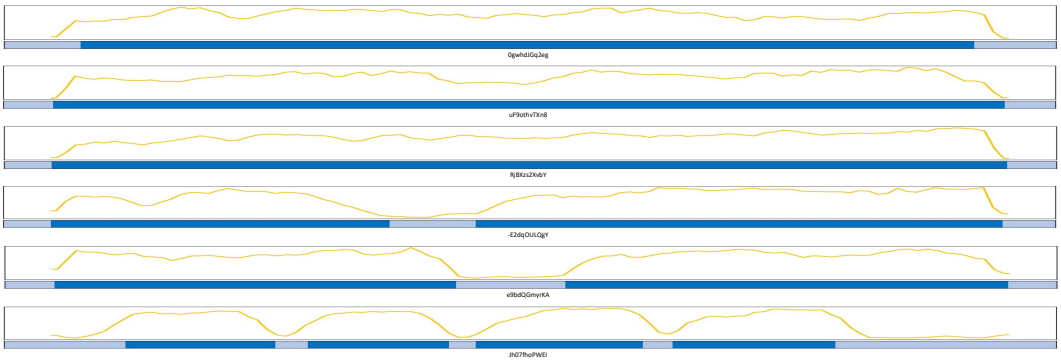

Figure 8: Qualitative results of the class activation sequences for ECM, on the ActivityNet v1.3 dataset.