This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Divide and Conquer for Single-frame Temporal Action Localization

Chen Ju[1], Peisen Zhao[1], Siheng Chen[1], Ya Zhang[1]✉, Yanfeng Wang[1], Qi Tian[2]
[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University [2]Huawei Cloud & AI
{ju_chen, pszhao, sihengc, ya_zhang, wangyanfeng}@sjtu.edu.cn, tian.qi1@huawei.com

Abstract

Single-frame temporal action localization (STAL) aims to localize actions in untrimmed videos with only one timestamp annotation for each action instance. Existing methods adopt the one-stage framework but couple the counting goal and the localization goal. This paper proposes a novel two-stage framework for the STAL task with the spirit of divide and conquer. The instance counting stage leverages the location supervision to determine the number of action instances and divide a whole video into multiple video clips, so that each video clip contains only one complete action instance; and the location estimation stage leverages the category supervision to localize the action instance in each video clip. To efficiently represent the action instance in each video clip, we introduce the proposal-based representation, and design a novel differentiable mask generator to enable the end-to-end training supervised by category labels. On THUMOS14, GTEA, and BEOID datasets, our method outperforms state-of-the-art methods by 3.5%, 2.7%, 4.8% mAP on average. And extensive experiments verify the effectiveness of our method.

1. Introduction

Temporal action localization (TAL) plays an important role in video understanding [35, 45, 38]. Its goal is to detect and classify all action instances in untrimmed videos. Recently, the fully-supervised setting [4, 18, 16, 34, 6, 14, 49] which requires frame-level supervision, has achieved impressive results; however, it is time-consuming and expensive to densely annotate each frame. On the other hand, the video-level weakly-supervised setting [19, 27, 28, 33, 20, 31] only needs the action category label of the whole video for localization. But lacking explicit location supervision fundamentally limits its empirical performance. To bridge the gap between fully-supervised and video-level weakly-supervised settings, a single-frame weakly-supervised TAL (STAL) is recently introduced [21], where a single frame (seedframe) is annotated for each action instance. STAL provides limited, yet precise action location supervision, and shows the potential to achieve great empirical performance and maintain cheap annotation overhead at the same time. This work explores this new STAL task.

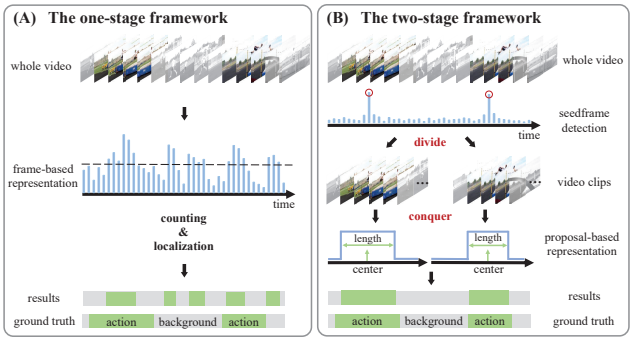

Figure 1. Comparison. **(A):** The one-stage framework couples the counting goal and the localization goal via thresholding, causing inferior localization results. **(B):** The two-stage framework detects seedframes to divide a whole video into multiple video clips, each of which contains only one complete action instance; then, it separately localizes the action instance in each video clip.

The existing STAL method [21] considers a one-stage framework, similar to video-level weakly-supervised methods [28, 11, 19]. Based on Multiple Instance Learning, this framework directly estimates the action probability at each individual frame; and then, by thresholding the action probability sequence, the framework simultaneously determines the number of action instances (counting) and localizes each action instance (localization); see Figure 1 (A). Since this one-stage framework couples the counting goal and the localization goal via thresholding, adjusting such a threshold empirically would highly affect both counting and localization performances, causing a serious coupling issue. Even tuning a threshold to provide the perfect counting results, this single and unified threshold might not be able to precisely localize all the action instances because each action instance could have its local sensitivity.

To solve the coupling issue, this work introduces a strategy of **divide and conquer** to decouple the counting goal and the localization goal. In other words, we aim to strategically divide the STAL task into multiple sub-tasks, each of which only needs to localize one action instance in a video

clip, then conquer each sub-task separately. Accordingly, we propose a novel two-stage framework, including the instance counting stage, which aims to determine the number of action instances and divide the whole video into several video clips, so that each clip contains only one complete action instance; and the location estimation stage, which aims to conquer each sub-task, *i.e.*, localize the time interval of the action instance in each video clip; see Figure 1 (B). The intuition of considering two stages is to separately exploit the location supervision for the instance counting stage, and the category supervision for the location estimation stage, from the limited single-frame supervision.

In the instance counting stage, we propose a seedframe detector to detect all the seedframes based on the location supervision. Since each seedframe indicates a unique action instance, the number of seedframes can reflect the total number of action instances. As a result, the detected seedframes guide us to divide a whole video into several video clips, each of which contains only one complete action instance. Next, in the location estimation stage, for each video clip, we localize the action instance with the category supervision, *i.e.*, adjust the location of the action instance so that the quality of action classification can be improved. Note that in this stage, since each video clip is supposed to cover a unique action instance, the single-frame weak supervision degenerates into the video-level weak supervision, and only handles simple single-instance localization.

To represent the location of the action instance in the video clip, existing video-level weakly-supervised methods mostly follow the frame-based representation [26, 19, 28], which directly estimates the action probability at each individual frame. However, the lack of precise frame-level supervision makes this representation inevitably suffer from a large solution space, resulting in two main issues: high false positives and lots of sparse and spiky actions. To avoid these issues, we introduce a more efficient proposal-based representation, which represents the action location with a gate-shaped proposal parameterized by the center and the length. It has two distinct advantages: i) its parameterization uses only two degrees of freedom for each action instance, which greatly reduces the solution space; ii) it naturally represents a time interval, promoting temporal smoothness and ruling out sparse and spiky actions. To adjust the center and the length of the action proposal via the category supervision, we aim to aggregate the frames within the proposal and specifically extract action-related features for action classification. Intuitively, a better estimation of the center and the length leads to better classification. To make this process trainable, we design a novel mask generator to transform the center and the length into a differentiable temporal mask, which indicates a time interval. Then, supervised by action category labels, we can adjust the center and the length of the proposal in an end-to-end fashion.

On three benchmark datasets, BEOID [5], GTEA [13], and THUMOS14 [7], the experimental results show that our method improves the average performance by 4.8%, 2.7%, 3.5% over the state-of-the-art methods. We further perform extensive ablation studies to reveal the effectiveness of each component, both quantitatively and qualitatively.

To summarize, our contributions include:

• We propose a novel two-stage framework for the STAL task with the spirit of divide and conquer. The instance counting stage leverages the location supervision to determine the number of action instances and divide a whole video into multiple video clips; and the location estimation stage leverages the category supervision to localize the action instance in each video clip.

• We adopt a proposal-based representation in the location estimation stage, which parameterizes the time interval of an action instance by the center and the length. To enable the end-to-end training supervised by category labels, we design a novel differentiable mask generator to transform the center and the length into a temporal mask.

• We conduct extensive experiments to validate the proposed method, which significantly outperforms the existing single-frame weakly-supervised method.

2. Related Work

Fully-supervised temporal action localization, which requires precise frame-level annotations, has made great progress. There are two popular output representations. The proposal-based representation [34, 32, 6, 4, 17, 41, 42] localizes the action instance with a proposal parameterized by the center frame and the preset length, then adjusts the proposal boundaries via a location regressor. The frame-based representation [49, 18, 16, 14, 48, 1, 42] trains a detector to search extreme frames (boundary or center frames), then combines extreme frames or estimates action lengths to produce final results. Both representations demand huge annotation costs, which are time-consuming and expensive. And without precise frame-level labels, the proposal-based representation is more effective due to its smaller solution space and temporal smoothness constraints. But it has not been used in weakly-supervised settings for two challenges: the lack of length labels and center labels, the uncertain number of action instances in the video. To the best of our knowledge, we are the first to explore the proposal-based representation in weakly-supervised settings.

Video-level weakly-supervised temporal action localization only requires cheap action category labels for training, thus reducing annotation costs. The existing methods mostly adopt the frame-based representation for action locations, and are divided into two branches. The MIL-based paradigm [39, 28, 19, 25, 33, 8, 20, 47] first trains a video classifier, then obtains frame-level action probabilities (representation) by calculating the Class Activation Sequence.

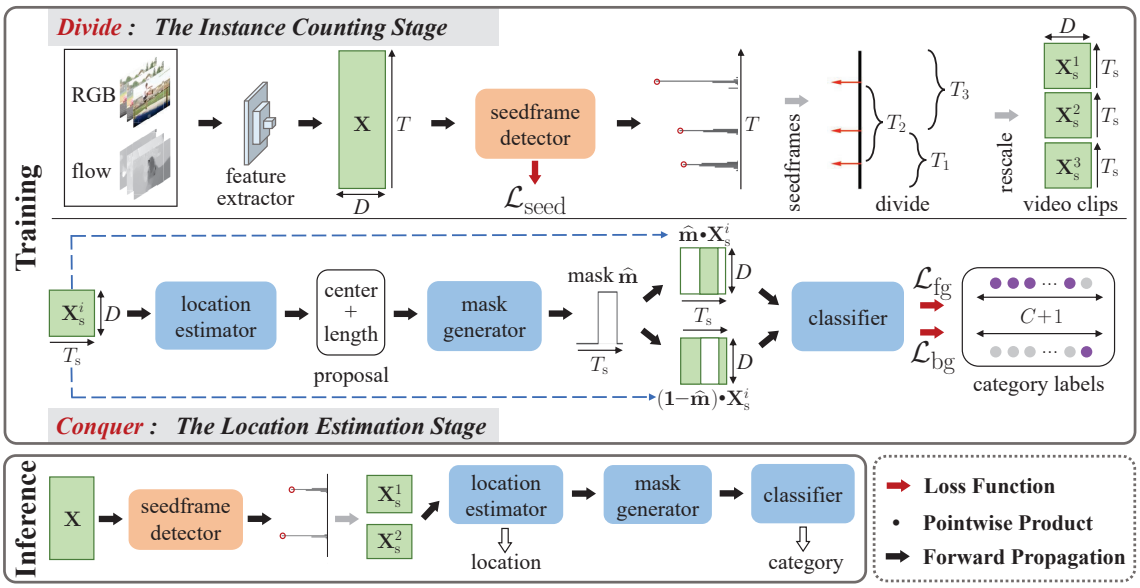

Figure 2. A two-stage framework with the spirit of divide and conquer. **In the instance counting stage**, we train a seedframe detector supervised by the location information of single-frame labels. Based on these detected seedframes, we divide the whole video into several video clips, promoting each video clip contains only one complete action instance. **In the location estimation stage**, given each video clip, a location estimator outputs a proposal parameterized by the center and the length, then a differentiable mask generator transforms the proposal into a temporal mask. We utilize the mask for temporal pooling to generate clip-level features, and classify these clip-level features into action categories supervised by the category information of single-frame labels.

The attention-based paradigm [27, 26, 20, 46] directly estimates frame-level action probabilities (representation) from raw data, which are regarded as attention to facilitate video classification. Besides, STARN [43] and 3C-Net [25] also explored the action frequency to provide more supervision. Nevertheless, all these methods individually estimate the action probability at each frame, and use empirical thresholds to produce localization results. Due to the lack of precise frame-level supervision, they are all troubled by serious false positives and trivial action fragments [19, 27, 11], causing inferior and impractical performances.

Single-point (frame) weak supervision is used to balance annotation costs and model performance. In semantic segmentation, WTP [2] first introduced this setting by annotating one point for each instance. And then, PDML [29] proposed metric learning between these single-point labels. In object counting, CLPS [10] designed the novel split-level loss and the false-positive loss. In spatial-temporal localization, SPOT [22] annotated one spatial location per-frame for each instance. In action recognition, ARST [24] annotated one video frame for each instance. Inspired by these methods, SF-Net [21] proposed the single-frame temporal action localization task (STAL). It used the one-stage framework to estimate frame-level action probabilities and produce final results by empirical thresholds. However, this one-stage framework couples the counting goal and the localization goal. On the contrary, we propose a novel two-stage framework to divide and conquer the STAL task.

3. Divide and Conquer

3.1. Problem Formulation

For a T-frame video, its feature is pre-extracted and denoted as $\mathbf{X} \in \mathbb{R}^{T\times D}$, where D is the feature dimension. Let $\mathcal{Y} = \{(\mathbf{y}_i, s_i, e_i)\}_{i=1}^{M}$ be all M action instances in the video, where $\mathbf{y}_i \in \mathbb{R}^C$ is the category label indicating C action categories; $s_i \in \mathbb{R}$ and $e_i \in \mathbb{R}$ are the start time and the end time. Temporal action localization (TAL) aims to design a model that detects and classifies M action instances $\mathcal{Y}$ from the input feature $\mathbf{X}$, *i.e.*, counting and localization.

This work considers the single-frame weakly-supervised setting (STAL) as proposed in [21]. Concretely, for the i-th action instance in an untrimmed training video, $(\mathbf{y}_i, s_i, e_i)$, only one single frame $(\mathbf{y}_i, t_i)$ is labeled by human annotators, where $t_i \in [s_i, e_i]$ provides the location supervision and $\mathbf{y}_i$ provides the category supervision. For comparison, the fully-supervised setting provides $(\mathbf{y}_i, s_i, e_i)$ for each action instance, and the video-level weakly-supervised setting only provides the category label $\mathbf{y}$ for the whole video.

3.2. Motivation and Overview

With the spirit of **divide and conquer**, we aim to divide the STAL task for a whole video into multiple sub-tasks of video-level weakly-supervised TAL for video clips; and then, we conquer each sub-task separately. Since each video clip is supposed to include only one action instance, each sub-task considers detecting and classifying one action in-

stance in each clip. In this manner, we decouple the counting and localization goals of the STAL task. Accordingly, we propose a novel two-stage framework, including the instance counting stage and the location estimation stage; see an illustration in Figure 2. The intuition of considering two stages is to separately exploit the location supervision and the category supervision from the limited single-frame supervision, to achieve divide and conquer.

The goal of **the instance counting stage** is to **divide**, *i.e.*, to determine the number of action instances and divide a whole video. Supervised by the location labels, this stage is fed with a whole video, and outputs multiple video clips, each of which contains only one complete instance.

The goal of **the location estimation stage** is to **conquer**, *i.e.*, to detect the time interval of the action instance in each video clip. Supervised by the category labels, this stage is fed with a video clip, and outputs the action category probability. In other words, this stage converts an action localization problem to an action classification problem.

3.3. Instance Counting Stage

This stage aims to determine the number of action instances in a given video, and divide the whole video into multiple video clips, ensuring each clip contains only one complete action instance. The stage includes two modules: seedframe detector, which estimates seedframe heatmaps by single-frame location labels, and video clip generation, which divides the whole video based on the seedframes.

Seedframe Detector. In the STAL task, each action instance is manually annotated with one single-frame location label, thus this single frame can be regarded as the seedframe of the corresponding action instance. Although seedframes are sparse, they already provide sufficient location supervision to distinguish different action instances.

Hence, to indicate the number of action instances in a given video, we evaluate the seedframe probability for each frame by a seedframe detector. The detector is fed with the video feature $\mathbf{X}$ to estimate a seedframe heatmap $\widehat{\mathbf{k}} \in \mathbb{R}^T$, where T is the total number of frames. Each element in $\widehat{\mathbf{k}}$ is the probability that the corresponding frame belongs to the seedframe. For training labels, we use the single-frame annotations. That is, if a frame is the annotated frame, it is regarded as a positive sample; otherwise, it is treated as a negative sample. And following [18, 48, 16], we adopt the weighted cross-entropy loss to optimize the detector:

$$\mathcal{L}_{\text{seed}} = \frac{1}{T^+} \sum_{t \in \Omega^+} \mathcal{H}(k_t, \widehat{k}_t) + \frac{1}{T^-} \sum_{t \in \Omega^-} \mathcal{H}(k_t, \widehat{k}_t), \quad (1)$$

where $k_t \in \{0, 1\}$ and $\widehat{k}_t \in [0, 1]$ are the seedframe label and the estimated probability of the t-th timestamp, $\mathcal{H}$ denotes the regular cross-entropy loss, Ω^+ and Ω^- are the positive and negative sample sets, T^+ and T^- are the number of positive and negative samples, respectively.

Video Clip Generation. Given the estimated heatmap $\widehat{\mathbf{k}}$, we select the seedframes by mining the local maxima. For any frame, we regard it as a seedframe if its probability reaches a local peak in $\widehat{\mathbf{k}}$ and exceeds the threshold θ. These filtered frames are then sorted and grouped into a seedframe set $\mathcal{P} = \{p_j\}_{j=1}^{M_p}$, where $p_j \in \mathbb{R}$ is the timestamp of the j-th seedframe and M_p is the number of seedframes. Since each seedframe indicates a unique action instance, M_p naturally reflects the number of action instances in the video.

To decouple the counting goal and the localization goal of the STAL task, we aim to divide the whole video into multiple video clips, ensuring that each video clip contains only one complete action instance. Since seedframes can distinguish different action instances, we realize the division based on the seedframe set $\mathcal{P}$. Concretely, for the j-th seedframe with the timestamp p_j, the time interval of its corresponding video clip is set as $[p_{j-1} + 1, p_{j+1} - 1]$. To unify the length of video clips, we rescale each video clip to T_{s} frames, and denote the clip feature as $\mathbf{X}_{\text{s}} \in \mathbb{R}^{T_{\text{s}} \times D}$, where D is the feature dimension of each frame.

3.4. Location Estimation Stage

Given a video clip generated by the instance counting stage, the location estimation stage aims to localize the time interval of the action instance with category labels, which is a video-level weakly-supervised setting. This stage includes a location estimator, which estimates the center and the length to represent the location of the action instance, a mask generator, which transforms the estimated center and length to a temporal mask, a feature aggregator, which leverages the temporal mask to pool action-related features, and a classifier, which classifies action-related features.

Location Estimator. To efficiently represent the time interval of the action instance, we consider the proposal-based representation, *i.e.*, a proposal parameterized by the center and the length. As we are not sure whether the seedframe in the video clip is at the action center, we form each proposal by the action length and the offset between the seedframe and the action center. Formally, we feed the feature $\mathbf{X}_{\text{s}}$ of the video clip into a location estimator, and produce the proposal $\mathbf{v} = (\Delta p + p, \ell)$, where $p \in \mathbb{R}$, $\Delta p \in \mathbb{R}$, and $\ell \in \mathbb{R}$ are the timestamp of the seedframe, the center offset, and the action length, respectively. Hence, the start time $\widehat{s}$ and the end time $\widehat{e}$ of the proposal are given by:

$$\widehat{s} = \Delta p + p - \frac{\ell}{2}, \quad \widehat{e} = \Delta p + p + \frac{\ell}{2}. \quad (2)$$

Finally, the time interval of the proposal is $\Psi = [\widehat{s}, \widehat{e}]$.

Mask Generator. To adjust the center and the length of the proposal by category supervision, we can aggregate the frames within the time interval of the proposal, and specifically extract only action-related features for classification. To make this process end-to-end trainable, we need to de-

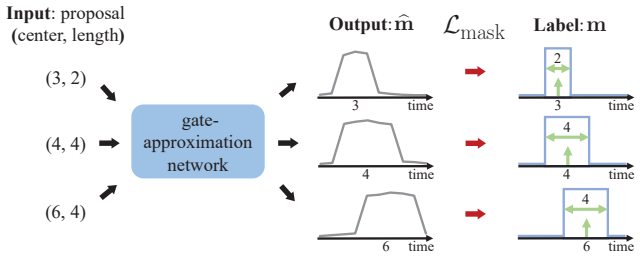

Figure 3. Paired training data of the gate-approximation network, which is simulated through Eq. (3). The input is a two-dimensional proposal, indicating the center and the length. The label is a T_{s}-dimensional gate-shaped mask, indicating the time interval.

sign a differentiable mask generator to transform the center and the length, $(\Delta p+p, \ell)$, into a temporal mask, $\mathbf{m} \in \mathbb{R}^{T_s}$, which indicates the time interval of the proposal. The element m_t of the temporal mask at the t-th timestamp is:

$$m_t = \begin{cases} 1, & \text{if } t \in [\widehat{s}, \widehat{e}], \\ 0, & \text{if } t \notin [\widehat{s}, \widehat{e}]. \end{cases} \tag{3}$$

Although mathematically simple, such a direct transformation is non-continuous at the two action boundaries, which makes it non-differentiable and hence infeasible to back-propagate training errors for model optimization.

To solve the non-differentiable issue, we propose two solutions. The first is to approximate Eq. (3) by a learnable network, and the second is to replace the gate-shaped mask with a Gaussian-shaped mask. The empirical comparison in Table 5 shows that the learnable approximation solution is better than the Gaussian-shaped solution. To implement the learnable approximation solution, the gate-approximation network is trained independently of the location estimation stage. In other words, during the end-to-end training of the location estimation stage, we freeze the weights of the gate-approximation network, so that it works as a deterministic network to transform the two-dimensional proposal into the T_{s}-dimensional approximate gate-shaped mask.

To train this gate-approximation network, we need to randomly simulate enough paired training data. As demonstrates in Figure 3, the input data is a two-dimensional simulated proposal, representing the center and the length. For each simulated proposal, based on Eq. (3), we calculate the corresponding T_{s}-dimensional gate-shaped temporal mask as its ground-truth label. That is, we assign positive labels to all frames inside the proposal interval, and negative labels to all frames outside the interval. We also use the weighted cross-entropy loss to optimize the network:

$$\mathcal{L}_{\text{mask}} = \frac{1}{T_{\text{s}}^{+}} \sum_{t \in \Lambda^{+}} \mathcal{H}(m_t, \widehat{m}_t) + \frac{1}{T_{\text{s}}^{-}} \sum_{t \in \Lambda^{-}} \mathcal{H}(m_t, \widehat{m}_t), \tag{4}$$

where $m_t \in \{0, 1\}$ and $\widehat{m}_t \in [0, 1]$ are the mask label and the network output of the t-th timestamp, $\mathcal{H}$ denotes the regular cross-entropy loss, Λ^{+} and Λ^{-} are the positive and negative sample sets, T_s^{+} and T_s^{-} are the number of positive and negative samples, respectively.

Foreground/Background Feature Aggregator. Given the output temporal mask $\widehat{\mathbf{m}}$ of the mask generator, we use it to filter out all action-related features, then calculate the clip-level foreground action features by temporal pooling:

$$\mathbf{x}_{\text{fg}} = \frac{1}{T_{\text{s}}} \sum_{t=1}^{T_{\text{s}}} \widehat{m}_t \mathbf{x}_t \in \mathbb{R}^D, \tag{5}$$

where $\mathbf{x}_t \in \mathbb{R}^D$ is the feature of the video clip at the t-th timestamp, $\widehat{m}_t$ is the temporal mask of the t-th timestamp. Similarly, the complement temporal mask $\mathbf{1} - \widehat{\mathbf{m}}$ is used to calculate the clip-level background feature:

$$\mathbf{x}_{\text{bg}} = \frac{1}{T_{\text{s}}} \sum_{t=1}^{T_{\text{s}}} (1 - \widehat{m}_t) \mathbf{x}_t \in \mathbb{R}^D. \tag{6}$$

Classifier. Given the clip-level foreground feature $\mathbf{x}_{\text{fg}}$ and the clip-level background feature $\mathbf{x}_{\text{bg}}$, we use a classifier to classify them, so that the location estimation stage can be supervised by category labels. To better distinguish foreground actions from the background, we carry out action classification and background modeling [27, 11]. Formally, the classifier is fed with $\mathbf{x}_{\text{fg}}$ ($\mathbf{x}_{\text{bg}}$), then outputs the clip-level foreground classification probability $\widehat{\mathbf{y}}_{\text{fg}} \in \mathbb{R}^{C+1}$ (the clip-level background-aware probability $\widehat{\mathbf{y}}_{\text{bg}} \in \mathbb{R}^{C+1}$), where C is the total number of action categories and the additional one denotes the background category.

To optimize the classifier, we adopt the regular cross-entropy loss between the predicted classification probability and the corresponding ground-truth label:

$$\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{bg}} + \beta \mathcal{L}_{\text{fg}} = \mathcal{H}(\mathbf{y}_{\text{bg}}, \widehat{\mathbf{y}}_{\text{bg}}) + \beta \mathcal{H}(\mathbf{y}_{\text{fg}}, \widehat{\mathbf{y}}_{\text{fg}}), \tag{7}$$

where $\mathcal{H}$ is the regular cross-entropy loss, β is a trade-off hyperparameter, $\mathbf{y}_{\text{fg}} = [y^1, ..., y^C, 0]^{\text{T}} \in \mathbb{R}^{C+1}$ and $\mathbf{y}_{\text{bg}} = [0, ..., 0, 1]^{\text{T}} \in \mathbb{R}^{C+1}$ are the foreground classification label and the background-aware label, respectively.

3.5. Inference

At testing time, different from previous methods [21, 11, 31], our framework does not need complex post-processing, *e.g.*, non-maximum suppression. For a given video, we first use the instance counting stage to detect seedframes, then based on these seedframes, divide the whole video into multiple video clips, so that each video clip contains only one complete seedframe. For each clip, we feed it into the location estimation stage to generate a proposal, thus obtaining the time interval of the action instance. And the classifier is used to predict the action category of the proposal. Each proposal is scored with the seedframe probability.

Table 1. Comparison with the state-of-the-art methods on THUMOS14. In addition to manually annotated single-frame labels, we also use the simulated single-frame labels, which are sampling from the ground-truth boundary labels through a uniform distribution. Our results significantly surpass the competitors under two types of single-frame labels, revealing the effectiveness of our method. TS [36], UNT [39], and I3D [3] denote three different feature extractors. AVG denotes the average mAP at IoU thresholds 0.1:0.1:0.7.

Supervision		Method	Feature	mAP@IoU							AVG
				0.1	0.2	0.3	0.4	0.5	0.6	0.7	
Full		SSN [49]	TS	66.0	59.4	51.9	41.0	29.8	19.6	10.7	39.77
		BSN [18]	TS	-	-	53.5	45.0	36.9	28.4	20.0	-
		A2Net [44]	I3D	61.1	60.2	58.6	54.1	45.5	32.5	17.2	47.03
		BU-TAL [48]	I3D	58.2	56.8	53.9	50.7	45.4	38.0	28.5	47.36
		PGCN [30]	I3D	69.5	67.8	63.6	57.8	49.1	-	-	-
		SALAD [37]	I3D	73.3	70.7	65.7	57.0	44.6	-	-	-
		AFSD [15]	I3D	-	-	67.3	62.4	55.5	43.7	31.1	-
Weak Video-level		STPN [26]	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	27.01
		WTALC [28]	I3D	55.2	49.6	40.1	31.1	22.8	14.8	7.6	31.60
		CMCS [19]	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.37
		BM [27]	I3D	64.2	59.5	49.1	38.4	27.5	17.3	8.6	37.80
		BaSNet [11]	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.30
		TSCN [46]	I3D	63.4	57.6	47.8	37.7	28.7	19.4	10.2	37.83
		DGAM [31]	I3D	60.0	54.2	46.8	38.2	28.8	19.8	11.4	37.03
		A2CL [23]	I3D	61.2	56.1	48.1	39.0	30.1	19.2	10.6	37.76
Weak Single-frame	Uniform	ARST [24]	UNT	24.3	19.9	15.9	12.5	9.0	-	-	-
		SF-Net [21]	I3D	68.3	62.3	52.8	42.2	30.5	20.6	12.0	41.24
		Ours	I3D	**70.2**	**63.5**	**55.6**	**44.7**	**32.3**	**22.0**	**12.3**	**42.93**
	Manual	SF-Net [21]	I3D	71.0	63.4	53.2	40.7	29.3	18.4	9.6	40.80
		Ours	I3D	**72.8**	**64.9**	**58.1**	**46.4**	**34.5**	**21.8**	**11.9**	**44.34**

3.6. Discussion and Comparison

Compared to the existing STAL method [21], our method is novel from two aspects. First, the training framework is different. The existing method couples the counting goal and the localization goal in a one-stage framework, causing inferior solutions to both goals; while our method designs a two-stage framework to strategically divide the STAL task into many sub-tasks, then separately conquer each sub-task. Second, given a video clip containing a complete instance, the output representation is different. The existing method adopts the frame-based representation, causing serious false positives and trivial actions; while our method considers the efficient proposal-based representation, which has a smaller solution space and temporal smoothness constraints.

Compared to fully-supervised methods, our method considers a similar representation but is novel in terms of the supervision setting. The limited single-frame supervision pushes us to explore an original two-stage training framework, with the spirit of divide and conquer. We separately exploit location supervision and category supervision for these two stages. And a novel mask generator is further designed to optimize this representation with category labels.

4. Experimental Results

4.1. Datasets and Evaluation

We conduct experiments on the following three datasets. For the sake of fairness, we adopt the single-frame labels provided by SF-Net [21] during training.

THUMOS14 [7] contains 413 untrimmed sports videos, which belong to 20 action categories. Following the convention, we train on 200 validation videos and evaluate on 213 testing videos. There are total 3007 single-frame annotations available for training, and each video contains an average of 15 action instances. Besides, action lengths and video lengths vary widely, making this dataset particularly challenging. **BEOID** [5] covers 58 videos in 34 categories. Following [21, 24], we set the proportion of training and testing videos to 80-20%, and obtain 594 single-frame annotations. **GTEA** [13] records 7 fine-grained actions in the kitchen. There are 28 videos in total, divided into 21 videos for training and 7 videos for testing. Each training video contains 17.5 single-frame labels on average.

Evaluation Metrics. We follow the standard protocols to evaluate with mean Average Precision (mAP) under different intersection over union (IoU) thresholds. And a proposal is regarded as positive only if both IoU exceeds the set threshold and the category prediction is correct.

4.2. Implementation Details

Feature Extraction. Following previous literature [21, 28, 19], we first split each untrimmed video into multiple frames (snippets), then extract optical flow via TV-L1 algorithm [40]. The video length T is set to 2500, 360, and 128 on THUMOS14, BEOID, and GTEA. We adopt the classic two-stream I3D network [3] pre-trained on Kinetics [3]

Table 2. Comparison on GTEA and BEOID. On both datasets, our method achieves the state-of-the-art performance. AVG denotes the average mAP at IoU thresholds 0.1:0.1:0.7.

Dataset	Method	mAP@IoU				AVG
		0.1	0.3	0.5	0.7	
GTEA	SF [21]	50.0	35.6	21.6	17.7	30.5
	SFB [21]	52.9	34.9	17.2	11.0	28.0
	SFBA [21]	52.6	32.7	15.3	8.5	26.4
	SF-Net [21]	58.0	37.9	19.3	11.9	31.0
	Ours	**59.7**	**38.3**	**21.9**	**18.1**	**33.7**
BEOID	SF [21]	54.1	24.1	6.7	1.5	19.7
	SFB [21]	57.2	26.8	9.3	1.7	21.7
	SFBA [21]	62.9	36.1	12.2	2.2	27.1
	SF-Net [21]	62.9	40.6	16.7	3.5	30.1
	Ours	**63.2**	**46.8**	**20.9**	**5.8**	**34.9**

as the feature extractor. After obtaining RGB and flow features, we concatenate them along the feature dimension, and get a 2048-dimensional vector for each frame.

Parameter Settings. For all datasets, we optimize our method by Adam [9] with a learning rate of 10^{-4}. For the hyperparameter β in Eq. (7), we set it to 2 on GTEA, 1.25 on BEOID and THUMOS14. The threshold θ is set to 0, 0.01, and 0.15 on GTEA, BEOID, and THUMOS14. The length of video clips T_{s} is set to 128 on THUMOS14, 64 on BEOID, and 32 on GTEA. To separately train the gate-approximation network, we simulate 0.1 million paired data, then optimize by Adam with a learning rate of 10^{-5}. The specific network architectures and more details are reported in the supplementary material.

4.3. Comparison with state-of-the-art methods

Table 1 compares our method with current state-of-the-art methods on THUMOS14. In addition to manually annotated single-frame labels, SF-Net [21] also provides the simulated single-frame labels, which are sampled from the ground-truth boundary labels via a uniform distribution.

Under two types of the single-frame labels, our method achieves gratifying results and demonstrates the effectiveness. Notably, when using manually annotated labels, our method significantly outperforms the state-of-the-art STAL method [21] with a substantial gain of 3.5% average mAP, bridging the gap between single-frame supervision and full supervision by a large margin. Moreover, our method even surpasses several fully-supervised counterparts [49, 48, 18] at some low IoU thresholds. The main reason is that these fully-supervised methods utilize the weaker feature extractor [36] or the weaker classifier in [39]. And due to the lack of precise frame-level supervision, our performance drops significantly as the IoU threshold increases.

Table 2 quantitatively compares our method with previous methods on GTEA and BEOID. SF, SFB, and SFBA are three benchmark models designed in SF-Net [21]. On GTEA, our method achieves a new state-of-the-art performance, with a considerable improvement of 2.7% average mAP. On BEOID, our method surpasses the best competitor by 4.8% in terms of the average mAP.

Table 3. Evaluation of divide and conquer on THUMOS14. Comparing (B) to (A), dividing the STAL task into multiple sub-tasks by a two-stage framework brings a significant improvement. Comparing (C) to (B), the proposal-based representation outperforms the frame-based representation in the location estimation stage.

ID	Division	Representation	mAP@IoU			AVG
			0.3	0.5	0.7	
(A)	no	frame-based	51.7	29.3	9.2	39.6
(B)	yes	frame-based	55.2	30.7	9.8	41.7
(C)	yes	proposal-based	**58.1**	**34.5**	**11.9**	**44.3**

Table 4. Ablation studies of the location estimation stage on THUMOS14. Δp is the center offset, $\mathcal{L}_{\mathrm{fg}}$ and $\mathcal{L}_{\mathrm{bg}}$ are the foreground classification loss and the background-aware loss in Eq. (7). AVG is the average mAP at IoU thresholds 0.1:0.1:0.7. All components are effective and essential to achieve the best performance.

$\mathcal{L}_{\mathrm{fg}}$	$\mathcal{L}_{\mathrm{bg}}$	Δp	mAP@IoU			AVG
			0.3	0.5	0.7	
✓			51.9	27.2	8.0	38.7
✓	✓		57.1	33.8	11.5	43.8
✓		✓	53.0	28.1	8.7	39.6
✓	✓	✓	**58.1**	**34.5**	**11.9**	**44.3**

Table 5. Comparison of the mask generator. AVG denotes the average mAP at IoU thresholds 0.1:0.1:0.7. The 'Gate-approximation' network is superior to the 'Gaussian-shaped' mask.

Solution	mAP@IoU			AVG
	0.3	0.5	0.7	
Gaussian-shaped	56.8	31.5	10.6	42.7
Gate-approximation	**58.1**	**34.5**	**11.9**	**44.3**

4.4. Ablation Studies and Comparison

Effectiveness of divide and conquer. Table 3 evaluates the effectiveness of the instance counting stage and the location estimation stage. (A): The baseline is a traditional one-stage framework using the frame-based representation. Its optimization and post-processing setting are similar to [21]. (B): We add the instance counting stage to the baseline, thus turning the one-stage framework into the two-stage framework. That is, first divide the whole video into several video clips by detecting seedframes, ensuring each video clip only contains one complete seedframe; then use the frame-based representation to localize the action instance in each video clip. (C): Based on (B), replace the frame-based representation with the proposal-based representation.

Comparing (B) to (A), there yields a significant boost in performance, with a gain of 2.1% average mAP. This phenomenon indicates that dividing the STAL task by detecting

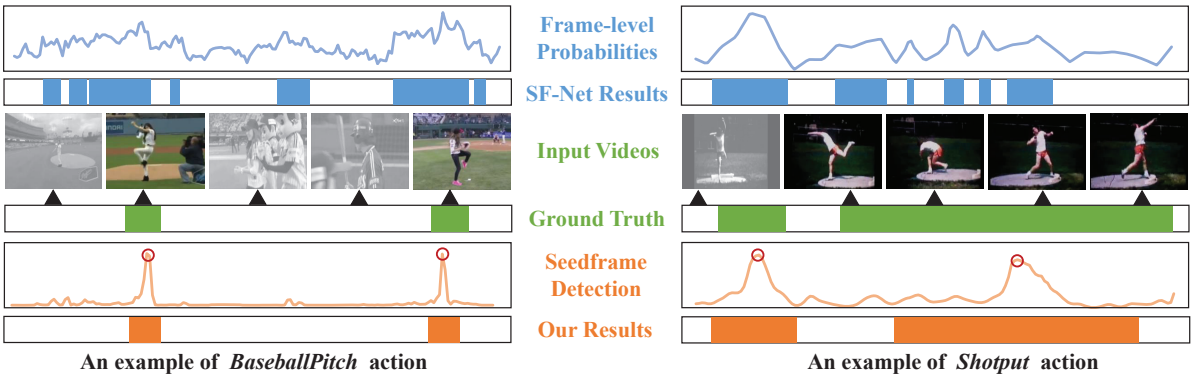

Figure 4. Qualitative comparison with SF-Net [21] on THUMOS14. The first two rows are estimated frame-level action probabilities and localization results of SF-Net. The middle two rows are input videos and ground-truth action intervals. The last two rows are estimated seedframe probabilities and localization results of our method. **Left:** Our method detects seedframes to indicate action instances, thus accurately determining the number of instances and ruling out false positives. **Right:** SF-Net suffers from non-smooth probabilities and obtains scattered action fragments. While our method localizes more complete actions by estimating the center and the length.

seedframes effectively decouples the counting and localization goals, thus simplifying the task difficulty. For further analysis, we collect some statistics based on the localization results. Under the IoU threshold 0.5, (B) wins a precision boost of 9.3% over (A), confirming that the divide-and-conquer two-stage framework significantly suppresses false positives. In terms of recall, (B) also obtains a gain of 3.6%, indicating that the divide-and-conquer framework can effectively reduce omissions and detect more complete action instances. Moreover, comparing (C) to (B), the proposal-based representation outperforms the frame-based representation by 2.6% average mAP, validating the effectiveness of the proposal-based representation.

Ablation studies of the location estimation stage. The foreground classification loss $\mathcal{L}_{\text{fg}}$, the background-aware loss $\mathcal{L}_{\text{bg}}$ and the center offset Δp are three important components. Table 4 investigates their contributions. (Without Δp, we treat the seedframe as the action center.)

Consistent with previous background modeling methods [27, 12, 11], the background-aware loss brings a considerable improvement, with a gain of 4.7% average mAP. The additional background category explicitly guides our method to distinguish actions from the background, resulting in more precise results. Surprisingly, the center offset only brings a slight gain of 0.8% in mAP@0.5. We conjecture that this is because the quality of action classification has a small account with the action center. As a result, category labels only provide limited guidance to adjust the action center. Nevertheless, all components are effective and essential to achieve the best performance.

Experimental comparison of the mask generator. Table 5 compares two solutions in the mask generator, *i.e.*, approximate the gate-shaped mask with the Gaussian-shaped mask or a learnable network. The gate-approximation network is superior to the Gaussian-shaped mask. It outputs a gate-approximation mask to assign equivalent weights for all action-related frames, which is more reasonable for action classification than Gaussian-shaped weights.

4.5. Qualitative Results

To intuitively demonstrate the superiority of our method, we visualize several results in Figure 4. We also reproduce the results of SF-Net [21] for comparison. As is evident, in both cases, the frame-level action probabilities of SF-Net have poor continuity or serious background noise, causing many false positives and trivial actions. On the contrary, by detecting seedframes, dividing the whole video into several video clips, and representing the action instance via a gate-shaped proposal, we decompose and simplify the STAL task, thus localizing more precise and complete actions.

5. Conclusions

This paper proposes a novel two-stage framework for STAL with the spirit of divide and conquer. The instance counting stage uses location labels to determine the number of action instances and divide a whole video into multiple video clips, so that each video clip contains only one complete instance; the location estimation stage uses category labels to localize the action instance in each video clip with the proposal-based representation. A novel mask generator is further designed to make this stage trainable. Extensive experiments on three benchmarks have verified the effectiveness and superior performance of our method.

Acknowledgements. This work is supported by the National Key Research and Development Program of China (No. 2019YFB1804304), National Natural Science Foundation of China (No. 61771306), SHEITC (No. 2018-RGZN-02046), 111 plan (No. BP0719010), and STCSM (No. 18DZ2270700), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

[1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–137. Springer, 2020.

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 549–565, 2016.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1130–1139, 2018.

[5] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio W Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 2, page 3, 2014.

[6] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3628–3636, 2017.

[7] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[8] Chen Ju, Peisen Zhao, Siheng Chen, Ya Zhang, Xiaoyun Zhang, and Qi Tian. Adaptive mutual supervision for weakly-supervised temporal action localization. *arXiv preprint arXiv:2104.02357*, 2021.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[10] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 547–562, 2018.

[11] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 11320–11327, 2020.

[12] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. Weakly-supervised temporal action localization by uncertainty modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[13] Peng Lei and Sinisa Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6742–6751, 2018.

[14] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 11499–11506, 2020.

[15] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3320–3329, 2021.

[16] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[17] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the ACM international conference on Multimedia (MM)*, pages 988–996, 2017.

[18] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[19] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1298–1307, 2019.

[20] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[21] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–437, 2020.

[22] Pascal Mettes, Jan C Van Gemert, and Cees GM Snoek. Spot on: Action localization from pointly-supervised proposals. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 437–453, 2016.

[23] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 283–299, 2020.

[24] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the*

IEEE International Conference on Computer Vision (ICCV), pages 8679–8687, 2019.

[26] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6752–6761, 2018.

[27] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[28] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[29] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 8843–8850, 2019.

[30] Mingkui Tan Yu Rong Peilin Zhao Junzhou Huang Runhao Zeng, Wenbing Huang and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[31] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1009–1019, 2020.

[32] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5734–5743, 2017.

[33] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018.

[34] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016.

[35] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4584, 2015.

[36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 2014.

[37] Guillaume Vaudaux-Ruth, Adrien Chan-Hon-Tong, and Catherine Achard. Salad: Self-assessment learning for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1269–1278, 2021.

[38] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7190–7198, 2018.

[39] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4334, 2017.

[40] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l 1 optical flow. In *Statistical and geometrical approaches to visual motion analysis*, pages 23–45. Springer, 2009.

[41] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5783–5792, 2017.

[42] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10156–10165, 2020.

[43] 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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[44] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing (T-IP)*, 29:8535–8548, 2020.

[45] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 982–990, 2016.

[46] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–54, 2020.

[47] 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 *Proceedings of the ACM international conference on Multimedia (MM)*, pages 738–746. ACM, 2019.

[48] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 539–555, 2020.

[49] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2914–2923, 2017.