RefineLoc: Iterative Refinement for Weakly-Supervised Action Localization

Alejandro Pardo[1]* Humam Alwassel[1]* Fabian Caba Heilbron[2] Ali Thabet[1] Bernard Ghanem[1]
[1]King Abdullah University of Science and Technology (KAUST) [2]Adobe Research
{alejandro.pardo,humam.alwassel,ali.thabet,bernard.ghanem}@kaust.edu.sa caba@adobe.com
http://humamalwassel.com/publication/refineloc

Abstract

Video action detectors are usually trained using datasets with fully-supervised temporal annotations. Building such datasets is an expensive task. To alleviate this problem, recent methods have tried to leverage weak labeling, where videos are untrimmed and only a video-level label is available. In this paper, we propose RefineLoc, a novel weakly-supervised temporal action localization method. RefineLoc uses an iterative refinement approach by estimating and training on snippet-level pseudo ground truth at every iteration. We show the benefit of this iterative approach and present an extensive analysis of five different pseudo ground truth generators. We show the effectiveness of our model on two standard action datasets, ActivityNet v1.2 and THUMOS14. RefineLoc shows competitive results with the state-of-the-art in weakly-supervised temporal localization. Additionally, our iterative refinement process is able to significantly improve the performance of two state-of-the-art methods, setting a new state-of-the-art on THUMOS14.

1. Introduction

Weak supervision has emerged as an effective way to train computer vision models using labels that are easy and cheap to acquire. This training strategy is particularly relevant for video tasks, where data collection and annotation costs are prohibitively expensive. In this paper, our goal is to localize actions in time when no information about the start and end times of these actions is available. The lack of temporal supervision makes it challenging to train models that discriminate between action and background segments. Recent methods for weakly-supervised temporal action localization focus on learning class activation maps using soft-attention [62], regularizing attention with an L1 loss [41], or leveraging co-activity and multiple instance learning losses [46]. Alternatively, other methods [36, 53] have focus on generating temporal boundaries using priors

*indicates equal contribution.

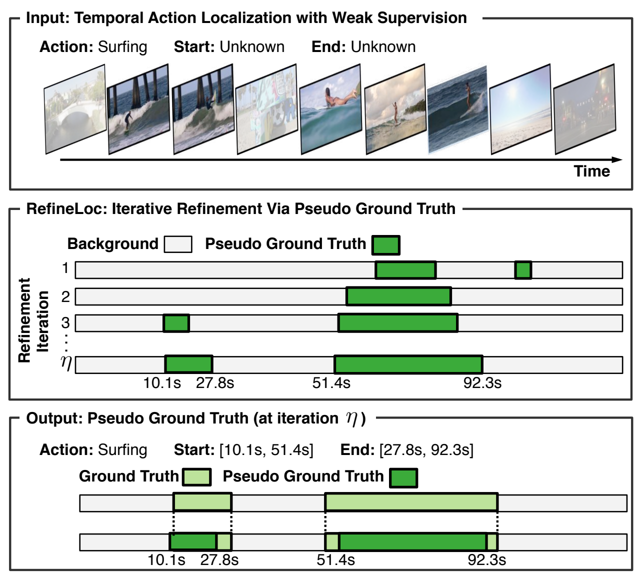

Figure 1: **Iterative Refinement for Weak Supervision.** We summarize the pseudo ground truth generation strategy used by RefineLoc. *Top*: The input is an untrimmed video, where only a video-level label (Surfing) is available; our goal is to correctly localize actions in time. *Middle*: RefineLoc aims to approximate the background-foreground labels through iteratively generating pseudo ground truth (dark green boxes) using information from a weakly-supervised model. Our key idea is to use the pseudo ground truth from iteration $\eta - 1$ to supervise the detection model at iteration η. *Bottom*: The pseudo ground truth of iteration η (dark green) closely approximates the actual ground truth (light green).

such as those encouraged by contrastive losses. All previous methods provide elegant strategies to localize actions in a weakly-supervised manner; however, they are all trained in a single shot and disregard all temporal cues. As a result, their performance lags far behind that of fully-supervised methods trained on temporal action annotations.

In the object detection domain, refining using pseudo ground truth considerably reduces the performance gap between fully and weakly-supervised object detectors [59, 71]. Such pseudo ground truth refers to a set of sampled object predictions from a weakly-supervised model, which are assumed as actual object locations in the next refinement iteration. However, these methods are not directly applicable to temporal action localization. We argue this is in part due to the lack of reliable *unsupervised* region proposals as in object detection.

In this paper, we propose RefineLoc, a weakly-supervised temporal localization method, which incorporates an iterative refinement strategy by leveraging pseudo ground truth. Figure 1 shows an example of the iterative refinement process RefineLoc employs via pseudo ground truth generation. Contrary to object detection methods, we build our refinement strategy to operate over snippet-level attention and classification modules, making it suitable for temporal localization.

The intuition behind our iterative refinement is to leverage a weakly-supervised model, which captures decent temporal cues about actions, to annotate snippets with pseudo foreground (action) and background (no action). This pseudo ground truth is then used to train a snippet-level attention module in a supervised manner. Although such pseudo labels are noisy, it has been shown that neural networks are reasonably robust against such label perturbations [50]. To avoid bias towards learning from easy examples, we randomly sample a subset of snippets for which we supervise with the pseudo labels. Our study of multiple pseudo ground truth generators shows that our simple model is competitive with the state-of-the-art. Furthermore, our iterative refinement process is generic and can be applied on top of more sophisticated models to further improve their performance.

Contributions: We summarize our contributions as 2-fold. **(1)** We introduce RefineLoc, an iterative refinement model for weakly-supervised temporal action localization. The model is crafted to leverage snippet-level pseudo ground truth to improve its performance over training iterations. **(2)** We show that RefineLoc's iterative refinement process improves the performance of two state-of-the-art methods, setting a new state-of-the-art on THUMOS14.[1]

2. Related Work

Action Recognition. The advent of action recognition datasets such as UCF-101 [58], Sports-1M [27], and Kinetics [28] has fueled the development of accurate action recognition models. Traditional approaches include extracting hand-crafted representations aimed at capturing spatiotemporal features [31, 61]; however, nowadays deep learning based approaches are more attractive due to their high capacity. For example, Simonyan and Zisserman [56] proposed to encode spatial and temporal information with Convolutional Neural Networks. Their two-stream model represents appearance with RGB frames and motion with stacked optical flow vectors. However, the two-stream model encodes each frame independently neglecting mid-level temporal information. To overcome this drawback, Wang *et al*. introduced the Temporal Segment Network (TSN) [63], an end-to-end framework that captures long-term temporal information. TSN along with other recent architectures (*e.g*. I3D [10] and C3D [60]) have become the *defacto* backbones for temporal action localization [47], action segmentation [19], and event captioning [65].

Fully-supervised Temporal Action Localization. Multiple strategies have been developed for temporal action localization with full-supervision available at training time [2, 14, 20, 22, 52, 68]. The first set of approaches used sliding windows combined with complex activity classifiers to detect actions [18, 43]. These methods paved the way for this type of research and established baselines and a reference for the difficulty of the problem. However, they manifested limitations regarding their run-time complexity. The second generation of methods used action proposals to speed up the search process [5, 6, 21, 34, 54]. These temporal proposals aim to narrow down the number of candidate segments the action classifier examines. A third generation of approaches learn action proposals and action classifiers jointly, while back-propagating through the video representation backbone [12, 64, 72]. Finally, recent methods make use of Graph Convolutional Networks by representing videos as graphs [67, 70]. Despite their significant performance improvements, all of these methods still rely on strong supervision that is prohibitively expensive to acquire.

Weakly-supervised Temporal Action Localization. The challenge in this task is to learn to discriminate between background and action segments without having explicit temporal training samples, but instead, only a coarse video-level label. The first methods proposed solutions consisting of hiding video regions to encourage their model to discover discriminative parts [30], and a soft-attention layer to focus on snippets that boost the video classification performance [62]. Similarly, [41] proposed an attention layer regularized with an L1 loss. Other works explored different alternatives such as co-activity loss combined with a multiple instance learning loss [46] and action proposal generation using contrast cues among action classification predictions [36, 53]. With the end goal of addressing the lack of temporal information, other works have innovated strategies such as incorporating temporal structure [69], modeling background [33, 42], using extra supervision (*e.g*. action count [39]), or single-frame label [37]. More recent methods have tried to reduce the supervision level by using self-supervised techniques [26]. Our work builds upon these ideas and complements them with a key insight: lever-

[1]To enable reproducibility and promote future research, we have released our source code and pretrained models on our project website.

aging pseudo labels while *iteratively* training the model.

Weak Supervision and Pseudo-labelling in Vision Tasks. Weak supervision has been widely studied in other vision tasks such as object detection [3, 44, 51, 57], semantic segmentation [45, 66], or other video tasks [17, 23, 24, 48]. For video tasks, a variety of weak supervision cues have been used including movie scripts [16, 29, 32, 38], action ordering priors [4, 11, 15, 49], and different levels of supervision [13]. These video related solutions have proposed innovative ways to reduce labelling expense; however, they still require laborious annotations (*e.g.* action spots) or privileged information (*e.g.* transcripts) that is difficult to obtain beyond a controlled setting. Concerning pseudo-labelling, it has been used to design state-of-the-art methods for weakly-supervised object detection [59, 71], train image classification backbones [8, 9], and build pose detectors [40]. These works have inspired our model, which addresses challenges unique to the weakly supervised temporal action localization task, namely the presence of only a sparse supervision signal (video-level action category) and of highly similar context surrounding the action [1].

3. RefineLoc

In this section, we discuss our RefineLoc architecture, the pseudo ground truth label generation, and the iterative refinement process. The input to our model is an untrimmed video and the expected output is a set of action segment predictions. RefineLoc is supervised on weak labels (*i.e.* video-level action labels) and **does not use** any temporal annotations of action instances. RefineLoc has two main components: a weakly-supervised temporal action localization (WSTAL) base model (Subsection 3.1) and an iterative refinement process (Subsection 3.2). Based on a trained WSTAL model, we generate pseudo background-foreground ground truth labels. We use these pseudo labels to supervise the training of a new WSTAL model. We repeat the process for η iterations to progressively improve the pseudo ground truth and refine the final action prediction segments. Figure 2 illustrates our approach.

3.1. WSTAL Base Model

The input to WSTAL is an untrimmed video, while the output is temporal action segment predictions. First, WSTAL extracts features form T non-overlapping snippets, which are then fed into both a snippet-level action classifier and a background-foreground attention module. Then, WSTAL combines the class activation and attention maps to produce a video label prediction $\hat{\mathbf{y}}$. During training, we supervise WSTAL with a cross-entropy loss between the ground truth video label $\mathbf{y}$ and the predicted label $\hat{\mathbf{y}}$. Finally, we post-process the learned class activation and attention maps to produce action segment predictions. In what follows, we discuss the details of each module in WSTAL.

Feature Extraction Module. To compare with other works, we use two feature extractor backbones: TSN [63] (pretrained by UntrimmedNets [62]) and I3D [10] (pretrained on Kinetics [28]). We split the input untrimmed video into T non-overlapping H-frame-long clip snippets (15 for TSN and 16 for I3D). We transform each snippet into a 2048-dimensional feature vector by concatenating the two 1024-dimensional activation vectors from the global pooling layer of each stream. Thus, this module outputs a $T \times 2048$ feature map $\mathbf{F}$.

Snippet-Level Classification Module. This module receives the feature map $\mathbf{F}$ and produces a $T \times N$ class activation map $\mathbf{C}$, where N is the number of action classes (100 classes in ActivityNet v1.2 [7] and 20 in THUMOS14 [25]). It consists of a multi-layer perceptron (MLP) with L Fully-Connected (FC) layers interleaved with ReLU activation functions. We reduce the size of each hidden layer by 2, which makes the last layer of size $\frac{2048}{2^{L-1}} \times N$.

Background-Foreground Attention Module. The objective of this module is to learn attention weights for each snippet to suppress the background snippets and to focus on foreground snippets. It transforms $\mathbf{F}$ into a $T \times 2$ background-foreground attention map $\mathbf{A}$. Similar to the Snippet-Level Classification Module, it consists of an MLP with L FC layers interleaved with ReLUs. Each hidden layer size is reduced by half, making the last FC layer of size $\frac{2048}{2^{L-1}} \times 2$. Other weakly-supervised action localization methods [36, 39, 41, 42, 53, 62] employ attention modules in their models. While we share a similar motivation, our attention module is different from theirs in one key aspect: their attention modules are only supervised by the video-level label for the purpose of improving the video classification, while our attention is supervised by both the video-level label and a set of pseudo background-foreground labels with the goal of improving the action segment localization. Subsection 3.2 details the pseudo ground truth label generation process. Unlike previous methods with a scalar attention, we model the attention explicitly with two values, one for foreground and one for background. We chose to do so because our method uses supervision directly on the attention values. Thus, instead of learning the attention with a logistic-regression loss, we learn it as a binary classification problem. We compare learning a scalar attention via logistic regression against our proposed two dimensional attention in the *supplementary material*.

Video Label Prediction Module. This module combines $\mathbf{C}$ and $\mathbf{A}$ to generate an N- dimensional probability vector $\hat{\mathbf{y}}$ for the video label. Specifically, we pass $\mathbf{C}$ through a softmax layer across the class dimension to get $\bar{\mathbf{C}}$ and pass $\mathbf{A}$ through two softmax layers. The first softmax layer operates across the background-foreground dimension to produce $\bar{\mathbf{A}}^{bf}$, while the second softmax layer operates across the time dimension (across snippets) of the foreground at-

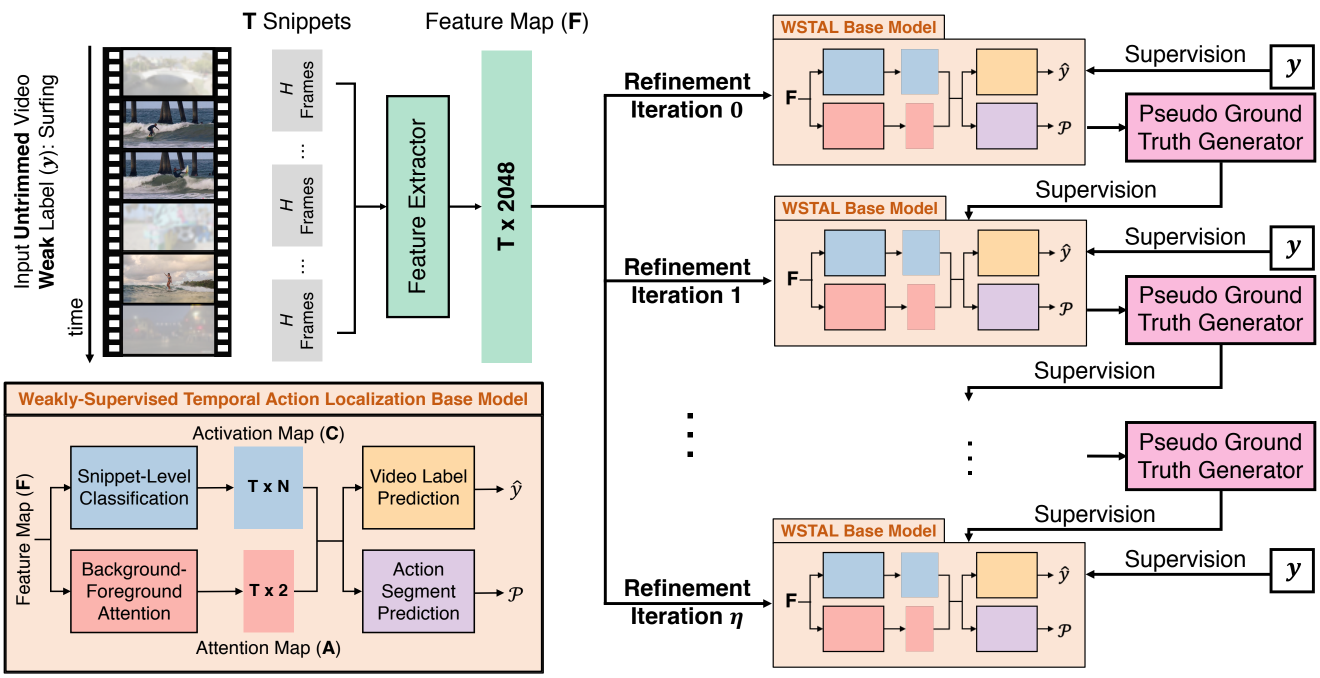

Figure 2: **Overview of RefineLoc Architecture.** Given an untrimmed video with only a weak label $\mathbf{y}$, we extract spatio-temporal feature map $\mathbf{F}$ from T non-overlapping H-frame-long snippets (*top left*). We feed $\mathbf{F}$ to an iterative refinement process (*right*). At each iteration, $\mathbf{F}$ passes through our WSTAL base model (*bottom left*) to compute a snippet-level activation map ($\mathbf{C}$) and a background-foreground attention map ($\mathbf{A}$). Both $\mathbf{A}$ and $\mathbf{C}$ are used to predict the video label $\hat{\mathbf{y}}$ and later to produce action segment predictions $\mathcal{P}$. At iteration 0, the pipeline is supervised using only $\mathbf{y}$. Subsequent iterations use both $\mathbf{y}$ and pseudo ground truth generated from the previous iteration. We stop the refinement process after η iterations.

tentions in $\bar{\mathbf{A}}^{bf}$ to produce $\bar{\mathbf{A}}^{time}$ as follows:

$$\bar{\mathbf{A}}^{bf}_{t,i} = \frac{\exp(\mathbf{A}_{t,i})}{\exp(\mathbf{A}_{t,1}) + \exp(\mathbf{A}_{t,2})}, \tag{1}$$

$$\bar{\mathbf{A}}^{time}_{t,i} = \frac{\exp(\bar{\mathbf{A}}^{bf}_{t,i})}{\sum_{t'=1}^{T} \exp(\bar{\mathbf{A}}^{bf}_{t',i})}. \tag{2}$$

Here, we use $\bar{\mathbf{A}}^{bf}$ as the network's predictions for the snippet-level background-foreground pseudo ground truth supervision (Subsection 3.2). Note that in Equations 1 and 2, $i = 1$ refers to background while $i = 2$ refers to foreground. Finally, this module computes the video-label prediction as $\hat{\mathbf{y}} = \sum_{t=1}^{T} (\bar{\mathbf{A}}^{time}_t \cdot \bar{\mathbf{C}}_t)$, where $\bar{\mathbf{A}}^{time}_t$ and $\bar{\mathbf{C}}_t$ are the foreground attention value and class activation vector of the t^{th} snippet. The video-label prediction uses a soft attention mechanism to emphasize the class activations of snippets with higher attention values.

Action Segment Prediction Module. This module post-processes $\bar{\mathbf{A}}^{bf}$ and $\bar{\mathbf{C}}$ to produce a set of action segment predictions $\mathcal{P}$. First, we filter out snippets for which the background attention value is greater than a threshold α_A. Then, we consider only the top-k classes in $\hat{\mathbf{y}}$. For each top class n, we filter out snippets that have classification score lower than a threshold α_C. Then, we generate contiguous segments by grouping snippets that are separated by at most one filtered-out (background) snippet. We do so to overcome noise in the filtering process and connect segments that are close to each other. This process can be done in other and more sophisticated ways, however we keep the simplicity of the base model and rely mainly on our iterative process. We assign to each predicted segment (t_1, t_2) the label n and the score s,

$$s = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} \left(\bar{\mathbf{A}}^{time}_t + \bar{\mathbf{C}}_{t,n} \right) + \hat{\mathbf{y}}_n. \tag{3}$$

where $\hat{\mathbf{y}}_n$ is the video-level predictions score for the n^{th} class. Note that each prediction that comes from the n^{th} top-k labels, has a different score s. Finally, to encode temporal context and deal with the ambiguity of action boundaries [1, 55], we inflate segments by 2 snippets at both ends.

3.2. Iterative Refinement Process

Let $\mathcal{M}_0$ be the WSTAL base model trained using the weak video labels only. We iteratively refine this base model and its action predictions by introducing supervision on the background-foreground attention module using snippet-level pseudo ground truth labels. Let $\mathcal{G}^{\mathcal{M}_\eta}$ be the pseudo ground truth generation function that uses information from $\mathcal{M}_\eta$ (the trained WSTAL base model after iteration η) to map each snippet to a pseudo background-foreground label. At iteration $\eta+1$, we train a new WSTAL base model $\mathcal{M}_{\eta+1}$ on the joint loss of the video-level label and the snippet-level pseudo ground truth labels from $\mathcal{G}^{\mathcal{M}_\eta}$. Specifically, we compute the loss for $\mathcal{M}_{\eta+1}$ on a given video in the following way,

$$\text{loss} = \mathcal{L}\left(\hat{\mathbf{y}}, \mathbf{y}\right) + \beta \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}\left(\bar{\mathbf{A}}_t^{bf}, \mathcal{G}^{\mathcal{M}_\eta}(t)\right), \quad (4)$$

where $\mathcal{L}$ is the cross-entropy loss and β is a trade-off coefficient to balance the loss signal of the pseudo ground truth with that of the video label. Note that the second cross-entropy loss is class-weighted to alleviate the imbalance in background and foreground pseudo labels.

Pseudo Ground Truth Generation. Intuitively, to obtain the maximum gain from the iterative refinement process, we want a pseudo ground truth generator that provides the closest approximation to the *true* snippet-level background-foreground ground truth labels, *i.e.* it should minimize the mislabeling rate. In order to overcome any possible bias learned by the pseudo ground truth generator and inspired by [30], we only fixate on a portion of the pseudo ground truth in a process we call *pseudo ground truth sampling*: at the start of each refinement iteration, we randomly sample a percentage S of snippets for which we apply the pseudo ground truth loss. We consider five different pseudo ground truth generation strategies and study their effects on the localization performance (Subsection 4.3).

(1) ***Uniformly Random Generator:*** This generator assigns a uniformly random pseudo label to each snippet.

(2) ***Distribution Aware Generator:*** This generator gives, with a biased probability, a random pseudo ground truth label to each snippet. The biased probability is equal to the average ratio of actual foreground to background snippets. This generator relies on information (namely the ratio) that requires access to strong temporal annotations. Thus, it does not align with the weakly-supervised setting, but we include it as a baseline reference only.

(3) ***Class Activation-Based Generator:*** This generator selects the pseudo ground truth label for a snippet t by thresholding its maximum class score, $\max(\bar{\mathbf{C}}_t)$.

(4) ***Attention-Based Generator:*** This generator produces the pseudo ground truth label for a snippet t by thresholding its foreground attention value, $\bar{\mathbf{A}}_t^{time}$.

(5) ***Segment Prediction-Based Generator:*** This generator assigns pseudo labels based on the set of prediction segments $\mathcal{P}$. A snippet is given a pseudo foreground label if it is covered by a segment prediction and a pseudo background label otherwise. We use this generator in our final model due to its attractive performance gain.

4. Experiments

4.1. Datasets and Evaluation Metric

We conduct our experiments on ActivityNet v1.2 [7] and THUMOS14 [25]. Both datasets consist of untrimmed videos with (weak) video-level action labels and have (strong) temporal annotations of action instances. However, *we discard the strong annotations during training*.

THUMOS14 [25]. This dataset has 1010 validation and 1574 testing videos annotated with 101 sport-related action classes at the video-level. Among these videos, only 200 validation and 213 testing videos have temporal annotations for 20 sport actions. As in prior work [20, 72], we only consider these 20 classes, use the 200 validation videos to train, and use the 213 testing videos to evaluate performance.

ActivityNet v1.2 [7]. This dataset has 9682 untrimmed videos annotated with 100 activity classes. It is split into training, validation, and testing subsets, where the testing subset labels are withheld for an annual challenge. Following other methods [46, 53], we use the training subset (4819 videos) to train and the validation subset (2383 videos) to test the performance. ActivityNet is a challenging dataset due to its large-scale nature and, unlike THUMOS14, its diverse classes ranging from household activities to sports.

Evaluation Metric. We compare methods according to mean Average Precision (mAP). We report mAP at multiple temporal Intersection-over-Union (tIoU) thresholds. We take the average mAP across tIoU thresholds 0.5:0.05:0.95 as the main metric for ActivityNet v1.2 and the mAP at tIoU threshold 0.5 as the evaluation metric for THUMOS14.

4.2. Implementation Details

We extract features from two different architectures: an I3D model [10], and the same pre-trained TSN [63] model used in AutoLoc [53], with 16 and 15 number of frames per snippet (H), respectively. We choose $L = 2$ layers for the snippet-level classification and background-foreground attention modules. In the action segment prediction module, we set (α_A, α_C) to $(0.5, 0.005)$ for ActivityNet and $(0.5, 0.35)$ for THUMOS14. We consider the top-2 labels when generating segment predictions in both datasets. At every iteration, we randomly sample $S = 80\%$ of the pseudo labels. Finally, we use an initial learning rate of 10^{-4} for ActivityNet and 10^{-3} for THUMOS14, and decay the learning rate by 0.9 when the validation loss saturates. We train for 50 epochs per refinement iteration and pick the best model with the lowest validation loss from Equation 4.

4.3. Ablation Study

In this subsection, we present multiple ablation studies motivating the design choices for our RefineLoc approach. First, we study the performance of several pseudo ground truth generators and the influence of the loss trade-off coef-

Pseudo Ground Truth Generator	β 0	1	2	4	8	16
Uniform Random	— 9.66 —	9.66	9.66	9.66	9.66	9.66
Distribution Aware		17.39	19.10	20.00	17.73	18.30
Class Activation		23.09	23.02	22.93	22.86	22.85
Attention		**23.15**	23.13	22.97	23.00	22.94
Segment Prediction		23.04	**23.15**	**<u>23.24</u>**	**23.11**	**23.09**

Table 1: **Effects of pseudo ground truth generator and loss trade-off coefficient β on ActivityNet v1.2**. The segment prediction-based generator with $\beta = 4$ shows the highest performance (underlined). Bold numbers mark the best performing generator for each β.

Refinement Iteration	0	1	2	3	4	5
RefineLoc	9.66	19.14	22.66	**23.24**	22.94	22.95

Table 2: **Effects of refinement**. We show the gain from our iterative refinement on ActivityNet v1.2. Note the significant improvement over iterations: 13.58% in 3 iterations.

ficient β (Equation 4) on the performance of each generator. Afterwards, we analyze how our model's performance changes from one refinement iteration to the next. Finally, we present a diagnosis study (using the DETAD [1] diagnostic tool) of the detection results before and after our iterative refinement process. We present all the studies in this subsection using ActivityNet v1.2 [7] dataset along with I3D features. For all the experiments in this section we report average mAP at tIoU thresholds 0.5:0.05:0.95. Refer to the *supplementary material* for the study results on ActivityNet v1.2 using TSN features as well as on THUMOS14 [25] using I3D and TSN features.

Effects of the Pseudo Ground Truth Generator and the Loss Trade-off Coefficient β. Table 1 summarizes the best average mAP performance using the five generators with five β values. The baseline model $\mathcal{M}_0$ ($\beta = 0$) achieves 9.66% average mAP at tIoU=0.5:0.05:0.95. We observe a performance improvement over $\mathcal{M}_0$ across all generator types and β values. This shows the effectiveness of our iterative refinement process. Moreover, we observe that the segment prediction-based generator is the best among the five generators. We hypothesize that this generator is better, since it has access to information from both the class activation and attention maps. Moreover, $\beta = 4$ strikes the best balance between the video label loss and the background-foreground pseudo ground truth loss. We observe similar results on THUMOS14: the best generator is the segment prediction-based one and the best β is 4.

Performance over Refinement Iterations. Table 2 shows the evolution of RefineLoc's performance across five refinement iterations. We obtain the highest performance (average mAP of 23.24%) after $\eta = 3$ iterations. This is a significant 13.64% increase over our baseline model $\mathcal{M}_0$ (iteration 0 in the table). We also see that refining $\mathcal{M}_0$ for a single iteration boosts the performance by 9.48%. This clearly shows the effectiveness of leveraging the pseudo ground truth labels during training. We observe similar results on THUMOS14: the best performance is achieved after $\eta = 3$ refinement iterations.

(a) Methods using TSN features

Method	0.5	0.75	0.95	Avg.
UntrimmedNets [62]	7.4	3.2	0.7	3.6
AutoLoc [53]	27.3	15.1	3.3	16.0
TSM [69]	28.3	17.0	3.5	17.1
CMCS [35]	33.9	19.9	5.1	20.5
CleanNet [36]	37.1	20.3	5.0	21.6
RefineLoc ($\eta = 0$)	25.8	11.5	2.8	13.3
RefineLoc ($\eta = 5$)	**38.8**	**22.2**	**5.3**	**23.2**

(b) Methods using I3D features

Method	0.5	0.75	0.95	Avg.
W-TALC [46]	37.0	-	-	18.0
3C-Net [39]	35.4	-	-	21.1
3C-Net† [39]	37.2	-	-	21.7
CMCS [35]	36.8	22.0	5.6	22.4
BaS-Net [33]	38.5	**24.2**	**5.6**	**24.3**
RefineLoc ($\eta = 0$)	19.2	8.0	2.3	9.7
RefineLoc ($\eta = 3$)	**38.7**	22.6	5.5	23.2

Table 3: **State-of-the-art Weak Supervision on ActivityNet v1.2**. RefineLoc outperforms other methods using TSN features (a) and is competitive using I3D features (b). 3C-Net† [39] uses number of instances per video as extra supervision.

(a) Methods using TSN features

Method	0.3	0.4	0.5	0.6	0.7
UntrimmedNets [62]	28.2	21.1	13.7	-	-
W-TALC [46]	32.0	26.0	18.8	-	6.2
CMCS [35]	37.5	29.1	19.9	12.3	6.0
AutoLoc [53]	35.8	29.0	19.9	12.3	6.0
CleanNet [36]	37.5	29.1	23.9	13.9	7.1
BaS-Net [33]	**42.8**	**34.7**	**25.1**	**17.1**	**9.3**
RefineLoc ($\eta = 0$)	7.0	4.2	2.9	1.3	0.6
RefineLoc ($\eta = 4$)	36.1	29.6	22.6	12.1	5.8

(b) Methods using I3D features

Method	0.3	0.4	0.5	0.6	0.7
W-TALC [46]	40.1	31.1	22.8	-	7.6
CMCS [35]	41.2	32.1	23.1	15.0	7.0
TSM [69]	39.5	-	24.5	-	7.1
3C-Net [46]	40.9	32.3	24.6	-	7.7
3C-Net† [46]	44.2	34.1	26.6	-	8.1
Nguyen *et al.* [42]	**46.6**	**37.5**	26.8	17.6	9.0
BaS-Net [33]	44.6	36.0	**27.0**	**18.6**	**10.4**
RefineLoc ($\eta = 0$)	34.8	27.7	19.5	10.7	4.60
RefineLoc ($\eta = 14$)	40.8	32.7	23.1	13.3	5.3

Table 4: **State-of-the-art Weak Supervision on THUMOS14**. RefineLoc is competitive using both feature types (tables **a** and **b**). † uses extra supervision from the number of instances per video.

(a) Generalizability of RefineLoc to other base models

Method	0.3	0.4	0.5	0.6	0.7
W-TALC Code [46]	42.98	34.59	26.99	**17.74**	**9.42**
W-TALC Code + **RefineLoc**	**44.10**	**35.08**	**27.66**	17.67	9.14

(b) Generalizability of RefineLoc to other base models

Method	0.3	0.4	0.5	0.6	0.7
BaS-Net Code [33]	43.40	35.16	26.26	18.59	10.16
BaS-Net Code + **RefineLoc**	**45.10**	**36.50**	**28.03**	**18.95**	**10.36**

Table 5: **Generalizability of RefineLoc.** Our iterative refinement process generalizes to base models: (a) W-TALC [46] and (b) BasNet [33]. We outperform W-TALC and BasNet baseline using 2 and 5 refinement iterations, respectively. By refining BasNet [33], we set a new state-of-the-art performance on THUMOS14.

4.4. State-of-the-Art Comparison and Generalizability

On ActivityNet v1.2 [7] (Table 3). RefineLoc with TSN features outperforms state-of-the-art, CleanNet [36], by 1.6% in average mAP (Table 3a), while RefineLoc with I3D features shows competitive performance to BaS-Net [33] (Table 3b). ActivityNet is large-scale and contains more diverse classes compared to THUMOS14. Thus, RefineLoc's strong performance on ActivityNet shows the effectiveness of our iterative refinement approach. We observe that our refinement process significantly enhances our base model, *i.e*. RefineLoc ($\eta = 0$), by 9.9% (TSN) and 13.5% (I3D) in average mAP.

On THUMOS14 [25] (Table 4). RefineLoc with TSN features (Table 4a) and with I3D features (Table 4b) exhibits competitive performance to state-of-the-art methods [33, 36, 42]. We observe that our refinement process significantly enhances our base model, *i.e*. RefineLoc ($\eta =$ 0), by 19.7% (TSN) and 3.6% (I3D) in mAP@tIoU$= 0.5$.

On Generalizability (Table 5). We chose our WSTAL-base model to be simple, compared to other state-of-the-art models, to highlight the main contribution of our work, *i.e*. the iterative refinement process. This process can lift the performance of such a simple model to compete and even outperform state-of-the-art methods on both datasets. Moreover, effectiveness of the refinement process is independent of the WSTAL-base model, which we demonstrate by generalizing our framework to other base models, namely W-TALC [46] and BaS-Net [33], on THUMOS14 using I3D features. These two methods employ attention-based models, where we apply our pseudo-background-foreground ground truth refinement process. Table 5 compares the results from the *released code* of the two methods vs. their performance after adding our iterative refinement process. By doing this, we significantly improve both base methods. In fact, BaS-Net is improved by 1.77% in mAP@tIoU$= 0.5$, setting a new state-of-the-art performance on THUMOS14 (28.03% mAP@tIoU$= 0.5$). Its important to note that the numbers obtained with the released codes differ from the ones reported in [33, 46].

We show that our method is simple, yet effective. We demonstrate that the key component of RefineLoc is the iterative process, showing its effectiveness regardless dataset, features, or base model. Despite its simplicity, RefineLoc outperforms all other methods using TSN features on ActivityNet, and beats the state-of-the-art when using BasNet and W-TALC as base models on THUMOS14.

4.5. Error Analysis and Qualitative Results

Diagnosing Detection Results. To analyze the merits of the proposed refinement strategy, we conduct a DETAD [1] false-positive analysis of RefineLoc at refinement iterations 0 and 3. We present the results in Figure 3. The false-positive profile analysis provides a fine-grained categorization of false-positive errors and summarizes the distribution of these errors over the top $5G$ model predictions, where G is the number of ground truth segments in the dataset. After refinement (right plot), we observe that RefineLoc generates more high-scoring true positive predictions (towards $1G$). Despite the reduction of background and localization errors, there is an increase in confusion errors. We explain this increase due to the simplicity of our initial classification module. Besides, the extra supervision generated by the pseudo-ground truth encourage the model to improve the localization but not directly the label prediction.

Qualitative Results. Figure 4 shows some RefineLoc qual-

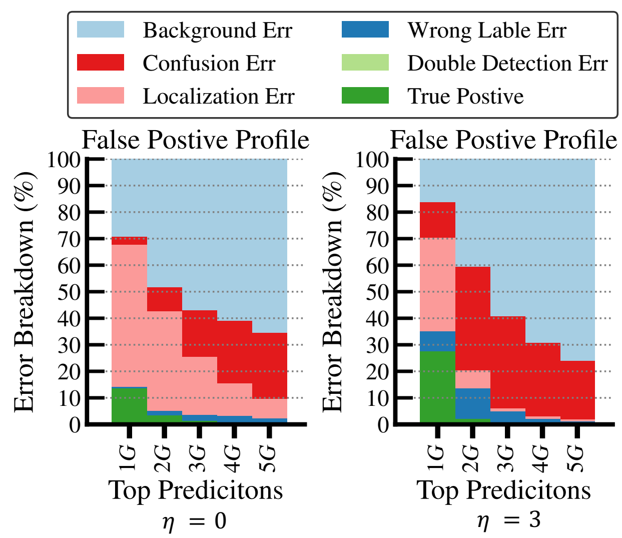

Figure 3: **Diagnosing Detection Results.** We present DETAD [1] false positive profiles of RefineLoc at refinement iterations 0 and 3. G represents the number of ground truth segments available in the ActivityNet dataset. Our refinement strategy clearly pushes more true positive predictions to the top $1G$ scoring predictions. RefineLoc also reduces background and localization error at later iterations, indicating temporally tighter predictions.

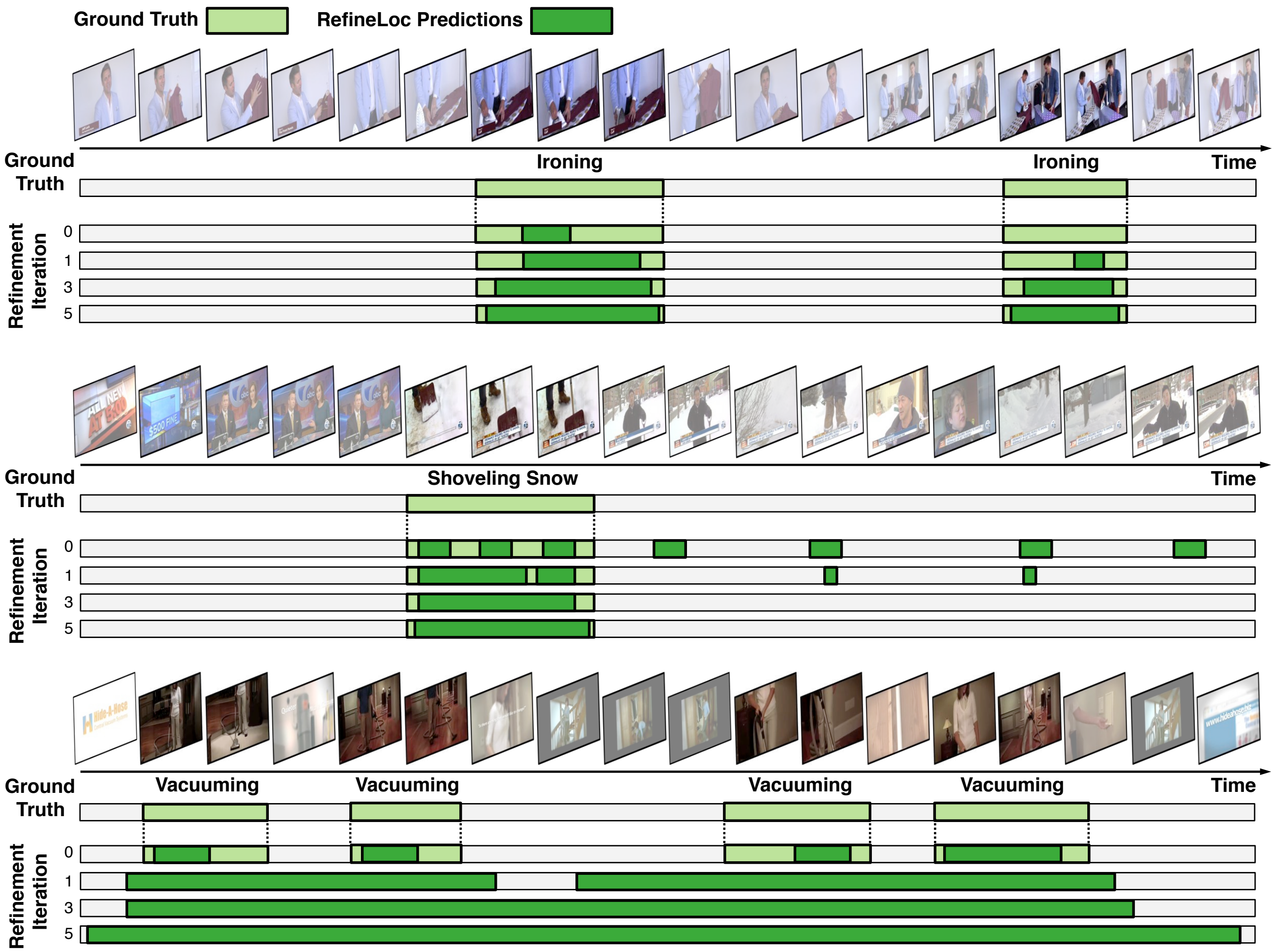

Figure 4: **Qualitative Results.** *Top*: RefineLoc successfully enhances prediction coverage and detects missed instances as iterations evolve. *Middle*: RefineLoc manages to merge disjoint predictions and remove wrong background predictions from one iteration to the next. *Bottom*: In the presence of large context, iterative refinement can hurt RefineLoc predictions, as visual similarity between foreground and background confuses our attention model.

itative detection results on ActivityNet. We present results for three different videos across different refinement iterations. The top video shows our method not only enhances its coverage over iterations, but is also able to detect a new instance at iteration 1 that was missed in the previous iteration. In the middle video, we see how RefineLoc manages to successfully merge different predictions over iterations. We also see erroneous predictions being cut off from iteration to iteration. The final example shows a failure case. Despite starting with decent predictions at iteration 0, our predictions diverge drastically in subsequent steps.

5. Conclusion

We have presented RefineLoc, a novel weakly-supervised temporal action localization method. RefineLoc uses an iterative refinement strategy, where snippet-level pseudo labels are generated and used at every training iteration. Our experiments have shown that RefineLoc is competitive with the state-of-the-art and that our general iterative refinement process boosts the results of other methods outperforming the state-of-the-art, suggesting that it could be used as an *off-the-shelf* strategy to refine results of future weakly-supervised methods for temporal action localization. As labelling videos for action localization is a massive time and cost bottleneck, RefineLoc takes a step closer to alleviating the need for these prohibitively expensive tasks.

Acknowledgments. This work is supported the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405.

References

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

[2] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. Action search: Spotting targets in videos and its application to temporal action localization. In *ECCV*, 2018.

[3] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016.

[4] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014.

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

[6] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, 2016.

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

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[9] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

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

[11] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D^3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*, 2019.

[12] 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*, 2018.

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

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

[15] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, 2018.

[16] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[17] Victor Escorcia, Cuong D Dao, Mihir Jain, Bernard Ghanem, and Cees Snoek. Guess where? actor-supervision for spatiotemporal action localization. *arXiv preprint arXiv:1804.01824*, 2018.

[18] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[19] J Gall and J Abu Farha. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019.

[20] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *ECCV*, 2018.

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

[22] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Victor Escorcia, Ranjay Khrisna, Shyamal Buch, and Cuong Duc Dao. The activitynet large-scale activity recognition challenge 2018 summary. *arXiv preprint arXiv:1808.03766*, 2018.

[23] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional videos. In *CVPR*, 2018.

[24] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016.

[25] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 2017.

[26] Mihir Jain, Amir Ghodrati, and Cees G. M. Snoek. Action-bytes: Learning from trimmed videos to localize actions. In *CVPR*, 2020.

[27] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[28] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[29] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 2017.

[30] 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*, 2017.

[31] Ivan Laptev. On space-time interest points. *IJCV*, 2005.

[32] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

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

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

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

[36] 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*, 2019.

[37] 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 *ECCV*, 2020.

[38] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *ICCV*, 2017.

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

[40] Natalia Neverova, James Thewlis, Riza Alp Guler, Iasonas Kokkinos, and Andrea Vedaldi. Slim densepose: Thrifty learning from sparse annotations and motion cues. In *CVPR*, 2019.

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

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

[43] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2014.

[44] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

[45] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015.

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

[47] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018.

[48] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017.

[49] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018.

[50] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[51] Miaojing Shi, Holger Caesar, and Vittorio Ferrari. Weakly supervised object localization using things and stuff transfer. In *ICCV*, 2017.

[52] 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*, 2017.

[53] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018.

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

[55] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017.

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

[57] 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*, 2017.

[58] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[59] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017.

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

[61] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

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

[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*, 2016.

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

[65] Huijuan Xu, Kun He, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Text-to-clip video retrieval with early fusion and re-captioning. In *AAAI*, 2019.

[66] Jia Xu, Alexander G Schwing, and Raquel Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015.

[67] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[68] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

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

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

[71] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2f: A weakly-supervised to fully-supervised framework for object detection. In *CVPR*, 2018.

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