Temporal Action Detection with Multi-level Supervision

Baifeng Shi
UC Berkeley

Qi Dai
Microsoft Research Asia

Judy Hoffman
Georgia Tech

Kate Saenko
Boston University & MIT-IBM Watson AI Lab

Trevor Darrell
UC Berkeley

Huijuan Xu
Penn State University

Abstract

Training temporal action detection in videos requires large amounts of labeled data, yet such annotation is expensive to collect. Incorporating unlabeled or weakly-labeled data to train action detection model could help reduce annotation cost. In this work, we first introduce the Semi-supervised Action Detection (SSAD) task with a mixture of labeled and unlabeled data and analyze different types of errors in the proposed SSAD baselines which are directly adapted from the semi-supervised classification literature. Identifying that the main source of error is action incompleteness (i.e., missing parts of actions), we alleviate it by designing an unsupervised foreground attention (UFA) module utilizing the conditional independence between foreground and background motion. Then we incorporate weakly-labeled data into SSAD and propose Omni-supervised Action Detection (OSAD) with three levels of supervision. To overcome the accompanying action-context confusion problem in OSAD baselines, an information bottleneck (IB) is designed to suppress the scene information in non-action frames while preserving the action information. We extensively benchmark against the baselines for SSAD and OSAD on our created data splits in THUMOS14 and ActivityNet1.2, and demonstrate the effectiveness of the proposed UFA and IB methods. Lastly, the benefit of our full OSAD-IB model under limited annotation budgets is shown by exploring the optimal annotation strategy for labeled, unlabeled and weakly-labeled data. [1]

1. Introduction

Temporal action detection is one of the most fundamental tasks in video understanding, which requires simultaneously classifying the actions in a video and localizing their start and end times. Recent success of temporal action detection models [55, 50, 18, 36, 4] highly relies on large

---
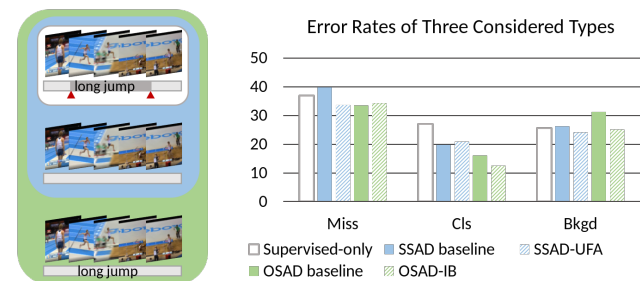
[1]https://github.com/bfshi/SSAD_OSAD.

Figure 1: *Left*: Data type for different tasks. FSAD uses only fully-labeled videos (white area). SSAD uses both fully-labeled and unlabeled videos (blue area). OSAD further uses weakly-labeled videos besides these two data types (green area). *Right*: Error analysis on SSAD/OSAD models. We consider three types of errors: 1) Action incompleteness (Miss), 2) Misclassification (Cls), and 3) Action-context confusion (Bkgd). Compared to the supervised-only model, the main type of error in SSAD baseline is Miss, while in OSAD baseline it is Bkgd. See Sec. 4.2.

amounts of fully-labeled training data with both classification and localization annotations. However, the annotation process, especially for localization, is extremely time-consuming and expensive. To alleviate this problem, one direction is to maximize the usage of unlabeled or weakly-labeled data to bring performance improvement with lower annotation cost. In this work, we study temporal action detection using fewer labeled videos together with other levels of supervision, e.g. unlabeled and weakly-labeled videos.

Learning from unlabeled data has been investigated in the task of semi-supervised image classification [28, 42, 1, 40] and shows promising results, while such problem setting is unexplored in the temporal action detection. We introduce the Semi-supervised Action Detection (SSAD) task and establish three SSAD baselines by incorporating three state-of-the-art Semi-Supervised Learning (SSL) models (Mean Teacher [42], MixMatch [1], FixMatch [40]) into a Fully-Supervised Action Detection (FSAD) backbone. For the purpose of **initial** evaluation on SSAD benchmark, we choose a straightforward yet effective FSAD method,

SSN [55], as our backbone and leave the development of more complex backbones for future work. However, directly applying SSL algorithms in the SSAD baselines only brings small improvement compared to the supervised-only model. To track down the main source of error, we conduct error analysis for the SSAD baselines (Fig. 1) and find the main problem of the SSAD baseline is **action incompleteness**, namely only detecting parts of the action. To help SSAD baseline better recognize actions, we borrow the idea from object-centric representations [8, 24, 49, 23] to extract more discriminative representation of action which is basically characterized by the foreground objects (humans).[2] There have been attempts to endow machines with the ability to detect salient moving objects. However, they either need manual annotation [44, 45, 41, 5], or make assumptions improper for action videos [52]. In this work, we propose to detect the foreground without supervision by leveraging the conditional independence between foreground and background motions, *i.e.*, the foreground motion is self-contained and not affected by the motion of background. Specifically, we learn the foreground attention by minimizing the conditional mutual information between foreground and background motion. To this end, our proposed unsupervised foreground attention (UFA) module successfully helps SSAD models recognize relatively complete actions without extra annotation cost.

Further, we consider weakly-labeled data with only video-level category labels which is in the middle of the cost-accuracy trade-off between fully-labeled and unlabeled data. It has been shown that weakly-supervised temporal action detection [26, 27, 35, 29, 19, 21] can save annotation cost without degrading the performance too much. Therefore, we further include weakly-labeled data into the SSAD model and form a unified framework with three levels of supervision, named Omni-supervised Action Detection (OSAD). As a baseline for OSAD, we simply add a video-level classification loss for the additional weakly-labeled data. However, training video-level classification to realize weak action localization may cause **action-context confusion** [19, 35], *i.e.*, the model is highly activated at non-action frames because they contain background *scene information* (*e.g.* swimming pool) which is highly indicative of the action category (*e.g.* swimming). This phenomenon is also verified in our error analysis (Fig. 1) where the OSAD baseline has higher action-context confusion than supervised-only model. To alleviate the issue, we propose an information bottleneck (IB) method to filter out the scene information extracted from non-action frames while preserving the action information by training action classification. Specifically, we regularize the entropy of non-

action frames which only contain scene information, thus reduce the action-context confusion(Fig. 1).

We conduct extensive experiments on SSAD and OSAD baselines, as well as the proposed UFA and IB. Moreover, we show the advantage of multi-level supervision over single-level supervision in a realistic scenario, where we search the best labeling policy under an annotation budget.

To sum up our main contributions, we: (i) propose the SSAD and OSAD tasks to utilize unlabeled and weakly-labeled data in temporal action detection, and establish several baseline models for them; (ii) design an unsupervised foreground attention module to alleviate the action incompleteness problem in SSAD baselines; (iii) design an information bottleneck method to solve the action-context confusion problem in OSAD baselines; (iv) validate the proposed SSAD and OSAD methods through extensive experiments, and show the advantage of our full OSAD-IB model under a realistic scenario where an annotation budget is given.

2. Related Work

Fully-Supervised Action Detection. The basic paradigms in FSAD methods share significant similarities with their counterparts in the object detection area [10, 9, 33, 32, 20]. The most common one is the two-stage pipeline, where proposals are first uniformly sampled from a video and then classified and re-localized. R-C3D [50] improves the proposal quality with a proposal subnet. SSN [55] optimizes a completeness loss to avoid the incomplete proposals. It also proposes the feature pyramid (STPP) to better capture the temporal structure, and the temporal actionness grouping (TAG) for proposal sampling. TAL-Net [4] borrows the architecture of Faster R-CNN [33] in object detection and makes it accommodate the action detection setting by multiple modifications. Recent works also focus on refining the temporal structures, *e.g.*, with multiple temporal scales [38], higher temporal resolution [36], precise temporal boundaries [18], or using graph networks to model the temporal relations [51, 54]. In this work, we choose SSN as the FSAD base model. Note that we discard STPP and TAG, and only keep the completeness loss to simplify the algorithm without hurting the performance too much.

Semi-Supervised Learning. Most semi-supervised learning methods can be summarized as training the model to predict the pseudo label or produce a consistent output of the input with different augmentations. Pseudo label (or self-training) methods [17] require to convert model outputs into hard pseudo labels using a sharpening function and encourage the model to predict the pseudo labels with high confidence. Consistency-based methods [16, 42, 28] exploit the output of themselves or their time-average version as "soft label," and make the model generate a consistent output when the input is randomly or adversari-

---

[2]In this paper we use the terms "background" and "foreground" to indicate the spatial regions in each frame, and "action" and "non-action" denote the temporal frames.

ally augmented. FixMatch [40] combines pseudo-label and consistency-based methods into a simple yet effective algorithm. There are other regularizations, *e.g.*, enforcing linearity [1] or minimizing entropy [11]. Some works apply weaker supervision in downstream areas [15, 22, 6, 34]. However, none of the SSL algorithms have been applied to the action detection problem. In this work, we try different types of SSL algorithms for action detection, including Mean Teacher [42], FixMatch [40], and MixMatch [1].

Object-Centric Action Understanding. Modern approaches for action recognition are built on top of deep models which take the whole frames as input to understand the action, *e.g.*, 2D ConvNets [14, 12], two-stream ConvNets [39, 48], and 3D ConvNets [46, 31, 3]. Recent studies show that the action recognition models relying on the static appearance or the background scene, may degrade the recognition performance [56] or lead to biased decisions [7]. Since action is mainly characterized by the movement of foreground objects, we expect the deep models to focus on the foreground motion for better recognition. Other works have also verified the superiority of object-centric representations [24, 23, 49]. To this end, in order to better utilize unlabeled data in our task, we propose the UFA module to detect foreground. Previous approaches to foreground detection usually need large amount of labeled data [44, 45, 41, 5]. A recent work learns the foreground detector based on the assumption that foreground motion and background motion in the same frame are mutually independent [52], which may not hold for action videos, considering that background motion may be affected by foreground objects through camera motion. We take the camera motion into consideration and fix the drawback in previous assumption [52] by checking the neighboring frames.

Weakly-Supervised Action Detection. WSAD learns to predict the actionness score (the likelihood of a proposal being an action) with only video-level classification label. Popular WSAD methods can be categorized as top-down or bottom-up methods. Top-down methods [19, 25, 29, 47] first train a video-level classifier and then obtain the proposal actionness score from the temporal class activation map (TCAM). Bottom-up methods [26, 27, 37, 53] directly predict the actionness score from raw proposals and learn to classify the video whose feature is given by averaging proposal features weighted by actionness score. In this work we adopt the bottom-up pipeline to train models on weakly-labeled data. However, it is known that WSAD methods are prone to recognize non-action frames as action (*i.e.* action-context confusion) when non-action frames also contain the category-indicative information. To address this issue, Liu *et al.* [19] attempt to separate action and context with hard negative mining by assuming that context clips should be stationary. Shi *et al.* [35] separate action and context by modeling the feature-level distribution with a generative

model. In this work, we design an information bottleneck to suppress the information in non-action frames.

3. Method

In temporal action detection, for a video $\mathbf{X}$ with T frames (either RGB or optical flow) $\mathbf{X} = (\mathbf{x}_t)_{t=1}^T$, we randomly sample N proposals $(s_i, e_i)_{i=1}^N$, where s_i and e_i are the start and end time of the i-th proposal. Normally, we utilize a trainable backbone g_θ to extract features for each frame, $\mathbf{z}_t = g_\theta(\mathbf{x}_t)$, and then obtain the feature $\mathbf{p}_i$ for each proposal by average pooling

$$\mathbf{p}_i = \frac{1}{e_i - s_i + 1} \sum_{t=s_i}^{e_i} \mathbf{z}_t. \quad (1)$$

For fully-labeled data, each proposal $\mathbf{p}_i$ has a class label $y_i \in \{0, 1, \cdots, C\}$, where C is the total number of action categories and $y_i = 0$ indicates a non-action proposal, and a regression score $r_i \in \mathbb{R}^2$ for the start and end time. For weakly-labeled data, we only have the video-level class label $y \in \{1, \cdots, C\}$. No label is available for unlabeled data. We denote the subsets of labeled, weakly-labeled, and unlabeled data by $\mathcal{S}$, $\mathcal{W}$, and $\mathcal{U}$, respectively. The basic pipeline under multi-level supervision is shown in Fig. 2.

3.1. Semi-Supervised Action Detection Baselines

We first build the SSAD baselines by integrating the Semi-Supervised Learning (SSL) algorithms into Fully-Supervised Action Detection (FSAD). We choose SSN [55] as the basic FSAD method. For SSL, we implement the state-of-the-art algorithms including Mean Teacher [42], MixMatch [1], and FixMatch [40] for our action detection setting. We briefly recap the aforementioned FSAD and SSL algorithms in the remainder of this section.

In SSN, we train a classification module h_{cls} and a regression module h_{reg} at proposal level, and in the meantime we also train a completeness module h_{comp} to predict the proposal completeness (denoted by $c_i \in \{0, 1\}$), indicating whether the proposal $\mathbf{p}_i$ is a complete action clip or not. A proposal is considered as incomplete ($c_i = 0$) if more than 80% of its own span is overlapped with an action clip, while its IoU with the clip is below 0.3. The total loss $\mathcal{L}^S$ for the i-th proposal consists of three parts:

$$\mathcal{L}^S(\mathbf{X}) = \mathcal{L}^S_{cls}(\mathbf{X}) + \alpha_c^S \mathcal{L}^S_{comp}(\mathbf{X}) + \alpha_r^S \mathcal{L}^S_{reg}(\mathbf{X}), \quad (2)$$

where each loss is given by

$$\mathcal{L}^S_{cls}(\mathbf{X}) = -\frac{1}{N} \sum_{i=1}^N \log h_{cls}(y_i | \mathbf{p}_i),$$

$$\mathcal{L}^S_{comp}(\mathbf{X}) = -\frac{1}{N} \sum_{i=1}^N \log h_{comp}(c_i | \mathbf{p}_i, y_i), \quad (3)$$

$$\mathcal{L}^S_{reg}(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N \|h_{reg}(\mathbf{p}_i, c_i) - r_i\|_1.$$
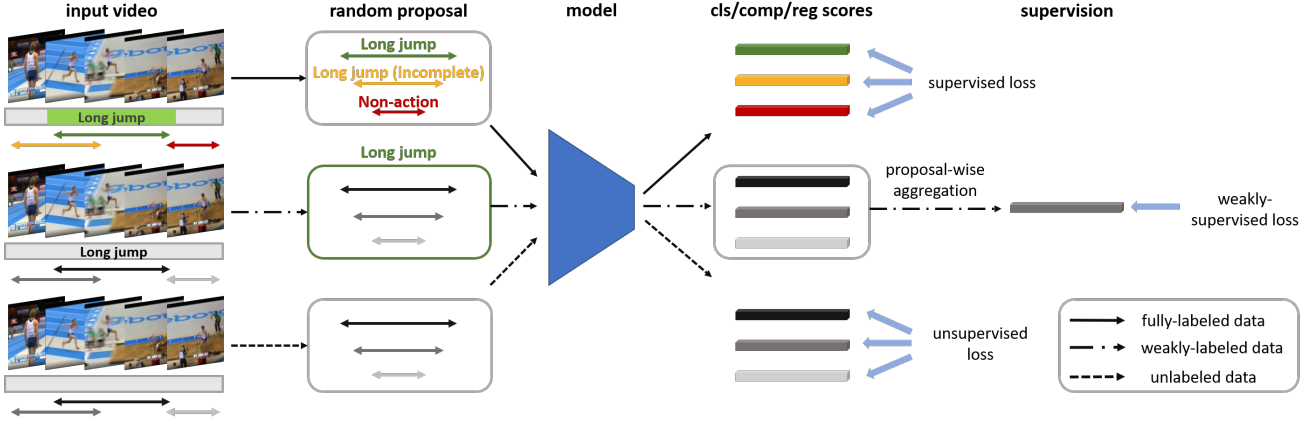
Figure 2: The basic pipeline of our proposed model for temporal action detection under multi-level supervision.

For unlabeled data, we adapt the unsupervised loss of each SSL algorithm for action detection problem:

$$\mathcal{L}^U(\mathbf{X}) = \mathcal{L}^U_{cls}(\mathbf{X}) + \alpha^U_c \mathcal{L}^U_{comp}(\mathbf{X}) + \alpha^U_r \mathcal{L}^U_{reg}(\mathbf{X}). \quad (4)$$

In the next, we introduce the specific design of $\mathcal{L}^U_*(\mathbf{X})$ ($* \in$ {"cls", "comp", "reg"}) in each SSL algorithm.

Mean Teacher optimizes the output consistency between two different augmentations of the same instance. It uses an Exponential Moving Average (EMA) of the backbone to extract features for one of the augmented input. The unsupervised loss is given by

$$\mathcal{L}^U_*(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \ell_*(h_*(\widetilde{\mathbf{p}}_i), h_*(\widetilde{\mathbf{p}}_i^{EMA})), \quad (5)$$

where $h_*(\cdot)$ indicates the output of corresponding module, $\widetilde{\mathbf{p}}_i$ is the feature of the augmented proposal, and the *EMA* superscript indicates the feature is extracted with the EMA version of the backbone. Note that $\widetilde{\mathbf{p}}_i$ and $\widetilde{\mathbf{p}}_i^{EMA}$ come from different inputs because the augmentation is stochastic. ℓ_* is the "distance" in the corresponding output space. We use KL divergence for classification and completeness scores, and L_1 distance for regression scores.

MixMatch enforces a linear output between input points. Following MixMatch method to deal with unlabeled data, we first obtain the pseudo label $\hat{h}_*(\mathbf{p}_i)$ for each proposal by sharpening the average output of K randomly augmented inputs, and then train the model on the mixed pseudo-labeled data with

$$\mathcal{L}^U_*(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \ell_*(h_*(\mathcal{M}_\lambda(\mathbf{p}_i, \mathbf{p}'_i)), \mathcal{M}_\lambda(\hat{h}_*(\mathbf{p}_i), \hat{h}_*(\mathbf{p}'_i))), \quad (6)$$

where $\mathbf{p}'_i$ is a proposal randomly sampled from the dataset, and $\mathcal{M}_\lambda(\cdot, \cdot)$ is the *mixup* function which is basically a linear interpolation with weights λ and $1 - \lambda$. λ is sampled from a Beta distribution [1].

FixMatch combines consistency-based and pseudo-label methods. It takes the augmented input and trains the model to predict the pseudo label $\hat{h}_*(\mathbf{p}_i)$:

$$\mathcal{L}^U_*(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(\max(\hat{h}_*(\mathbf{p}_i)) > \tau\right) \cdot \ell_*(h_*(\widetilde{\mathbf{p}}_i), \hat{h}_*(\mathbf{p}_i)). \quad (7)$$

The pseudo label is generated from the output of a *weakly*-augmented input. Note that we only train on pseudo labels with high confidence, *i.e.*, $\max(\hat{h}_*(\mathbf{p}_i)) > \tau$. Since this method does not apply to non-classification task such as regression, we do not add this term in $\mathcal{L}^U_{reg}$.

The overall objective for our proposed SSAD models is

$$\mathcal{L} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{X} \in \mathcal{S}} \mathcal{L}^S(\mathbf{X}) + \alpha^U \cdot \frac{1}{|\mathcal{U}|} \sum_{\mathbf{X} \in \mathcal{U}} \mathcal{L}^U(\mathbf{X}). \quad (8)$$

In all SSL algorithms, we exploit both spatial and temporal augmentations for video data. For spatial augmentations, we apply random noise and horizontal flip to all the frames in each proposal. We also design three temporal augmentations: (i) **Temporal Resampling**: In Eq. 1 we obtain the proposal feature by average pooling. In practice, we only sample L frames from the proposal, and take the average of their features as an efficient estimation [55]. In Temporal Resampling, we resample L frames from the proposal and take the new average as an augmented feature. (ii) **Temporal Resolution**: Instead of sampling L frames from each proposal, we sample $2L$ or $L/2$ frames. (iii) **Temporal Flip**: The video is played backwards. For the weak augmentation in Fixmatch, we employ only spatial augmentations without temporal ones. Please refer to supplementary for an evaluation of the augmentations.

3.2. Unsupervised Foreground Attention

The SSAD baseline is prone to miss part of the action (action incompleteness), as revealed by the error analysis in Fig. 1. Since actions are basically defined by the movement of foreground objects, our conjecture is that the model
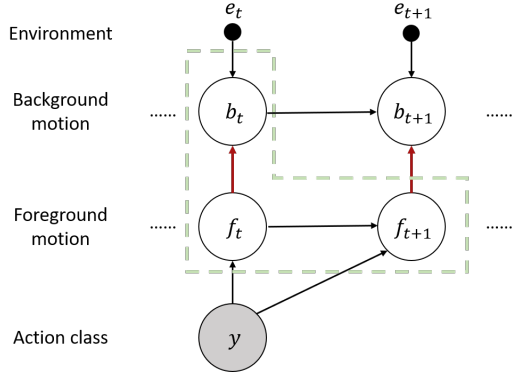
Figure 3: A (causal) graphical model for foreground/background motion in a video. Here we assume the motion is Markovian. The foreground motion f_t is determined by the action category y. The background motion b_t is affected by random factors e_t in the environment. Meanwhile, b_t is also affected by f_t through camera motion (red arrows). The structure in the green box indicates the conditional independence $(b_t \perp\!\!\!\perp f_{t+1})|f_t$.

can better recognize complete action clips by paying more attention to the foreground. Moreover, learning the attention without extra supervision can better utilize the unlabeled data. Therefore, we propose an unsupervised foreground attention (UFA) module to address the issue.

Intuitively, one can hardly estimate the motion of foreground from background, and vice versa, which indicates a certain level of "independence" between the **motions** of background and foreground. [52] assumes a complete independence between the foreground and background motions, which may not hold in action videos because of **camera motion**. An intuitive example is that we usually move the camera to track the foreground object when recording a video, which consequently causes movement of background scene. This creates a case that background motion is partially affected by foreground motion. Taking this into consideration, we describe the causal structure of foreground/background motions with the graph in Fig. 3. We denote the foreground and background motions in frame t by f_t and b_t. f_t is determined by the action category y, and b_t is affected by random factors e_t in the environment. f_t has an effect on b_t via camera motion. Although f_t and b_t are not completely independent, there is a conditional independence between b_t, f_t, and f_{t+1}, i.e., $I(b_t, f_{t+1}|f_t) = 0$, where I is the (conditional) mutual information. Intuitively, this means the foreground motion is a self-contained flow which is not affected by the background.

Suppose $\hat{\mathbf{z}}_t \in \mathbb{R}^{h \times w \times c}$ is the feature map before the final spatial pooling, i.e., $\mathbf{z}_t = \text{Pool}(\hat{\mathbf{z}}_t) \in \mathbb{R}^c$. Now we apply the UFA module $\text{Att}_\phi(\cdot)$ on $\hat{\mathbf{z}}_t$, i.e., $\mathbf{z}_t = \text{Pool}(\mathbf{a}_t * \hat{\mathbf{z}}_t)$, where $\mathbf{a}_t = \text{Att}_\phi(\hat{\mathbf{z}}_t) \in \mathbb{R}^{h \times w \times 1}$ and $*$ is the broadcastable

Hadamard product. We use a two-layer MLP for $\text{Att}_\phi(\cdot)$. To train UFA, we minimize the conditional mutual information:

$$\min_\phi I(\mathbf{B}_t, \mathbf{F}_{t+1}|\mathbf{F}_t)$$
$$\Leftrightarrow \quad \min_\phi H(\mathbf{F}_{t+1}|\mathbf{F}_t) - H(\mathbf{F}_{t+1}|\mathbf{F}_t, \mathbf{B}_t), \tag{9}$$

where H is the (conditional) entropy, and $\mathbf{F}_t$ and $\mathbf{B}_t$ are the random variables for foreground and background *motion* features. We denote their realizations by $\mathbf{f}_t$ and $\mathbf{b}_t$. When the input is optical flow, we directly extract motion feature from $\hat{\mathbf{z}}_t$, as the flow itself already represents motion:

$$\mathbf{f_t} = \text{Pool}(\mathbf{a}_t * \hat{\mathbf{z}}_t),$$
$$\mathbf{b_t} = \text{Pool}((1 - \mathbf{a}_t) * \hat{\mathbf{z}}_t). \tag{10}$$

When the input is RGB frame, we use the difference between two consecutive frame features as the motion feature

$$\mathbf{f_t} = \text{Pool}(\mathbf{a}_t * (\hat{\mathbf{z}}_{t+1} - \hat{\mathbf{z}}_t)),$$
$$\mathbf{b_t} = \text{Pool}((1 - \mathbf{a}_t) * (\hat{\mathbf{z}}_{t+1} - \hat{\mathbf{z}}_t)). \tag{11}$$

To optimize Eq. 9, we need an estimation of the entropy. If we assume the feature distributions are Gaussian, we have

$$H(\mathbf{F}_{t+1}|\mathbf{F}_t, \mathbf{B}_t) \propto \log E_{(\mathbf{f}_t, \mathbf{b}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - E(\mathbf{F}_{t+1}|\mathbf{f}_t, \mathbf{b}_t)\|_2^2,$$
$$H(\mathbf{F}_{t+1}|\mathbf{F}_t) \propto \log E_{(\mathbf{f}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - E(\mathbf{F}_{t+1}|\mathbf{f}_t)\|_2^2. \tag{12}$$

Therefore, we first train two predictors (two-layer MLP) u_ψ and u_ζ to approximate the conditional estimation:

$$\psi^* = \arg\min_\psi E_{(\mathbf{f}_t, \mathbf{b}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - u_\psi(\mathbf{f}_t, \mathbf{b}_t)\|_2^2,$$
$$\zeta^* = \arg\min_\zeta E_{(\mathbf{f}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - u_\zeta(\mathbf{f}_t)\|_2^2, \tag{13}$$

and then train UFA by optimizing

$$\min_\phi \log \frac{E_{(\mathbf{f}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - u_{\zeta^*}(\mathbf{f}_t)\|_2^2}{E_{(\mathbf{f}_t, \mathbf{b}_t, \mathbf{f}_{t+1})}\|\mathbf{f}_{t+1} - u_{\psi^*}(\mathbf{f}_t, \mathbf{b}_t)\|_2^2}. \tag{14}$$

In practice, we update ψ, ζ, and ϕ simultaneously to avoid the bi-level optimization. With our UFA module, we reduce the action incompleteness error and improve the performance of SSAD baselines for free (Sec. 4.3).

3.3. Omni-Supervised Action Detection with Information Bottleneck

Now we add weakly-labeled data to train the model with three levels of supervision, and form Omni-Supervised Action Detection (OSAD). For fully labeled and unlabeled data, we minimize $\mathcal{L}^S(\mathbf{X})$ and $\mathcal{L}^U(\mathbf{X})$ as introduced in Sec. 3.1. For weakly-labeled data, we optimize a video-level classification loss $\mathcal{L}^W(\mathbf{X})$ which is given by

$$\mathcal{L}^W(\mathbf{X}) = -\log h_{cls}(y|\mathbf{X}), \tag{15}$$

where $h_{cls}(y|\mathbf{X})$ is the video-level classification score given by the weighted average of the proposal classification scores

$$h_{cls}(y|\mathbf{X}) = \frac{\sum_{i=1}^{N} \lambda_i h_{cls}(y|\mathbf{p}_i)}{\sum_{i=1}^{N} \lambda_i}, \qquad (16)$$

where $\lambda_i = 1 - h_{cls}(y = 0|\mathbf{p}_i)$ is the probability of $\mathbf{p}_i$ being an action. Then the overall loss for OSAD baseline is

$$\mathcal{L} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{X} \in \mathcal{S}} \mathcal{L}^S(\mathbf{X}) + \alpha^U \frac{1}{|\mathcal{U}|} \sum_{\mathbf{X} \in \mathcal{U}} \mathcal{L}^U(\mathbf{X}) + \alpha^W \frac{1}{|\mathcal{W}|} \sum_{\mathbf{X} \in \mathcal{W}} \mathcal{L}^W(\mathbf{X}). \qquad (17)$$

However, **the OSAD baseline is prone to classify non-action frames as action frames (action-context confusion)**, as shown in Fig. 1. This issue is common when learning action detection from weakly-labeled data [19, 35]. Ideally, the recognition model is expected to classify the weakly-labeled videos based on action information (*e.g.* swimming) which could also benefit the action detection task. However, the model tends to take a "shortcut" instead and learns to classify action based on the scene information (*e.g.* the swimming pool), which would disrupt the detection by mistaking non-action frames with scenes for action frames. Now the question is, *how can we filter out the scene information and only keep the action information when only training the classification task*? Note that, although action frames contain both action and scene information, the non-action frames only contain the scene part. Thus, we propose to "unlearn" the scene information by penalizing all the information extracted from non-action frames.

Assuming the feature distribution to be Gaussian, the information of non-action frames can be estimated by

$$I \propto \log E_{\mathbf{X}} \left\{ \frac{1}{\sum_i \bar{\lambda}_i} \sum_i \bar{\lambda}_i \|\mathbf{p}_i - \frac{\sum_{i'} \bar{\lambda}_{i'} \mathbf{p}_{i'}}{\sum_{i'} \bar{\lambda}_{i'}} \|_2^2 \right\}, \qquad (18)$$

where $\bar{\lambda}_i = h_{cls}(y = 0|\mathbf{p}_i)$ is the likelihood of being a non-action proposal. Then we add this term into $\mathcal{L}^W$. Note that we also add a normalization term to avoid a trivial solution. The final loss on weakly-labeled data is

$$\mathcal{L}^W(\mathbf{X}) = -\log h_{cls}(y|\mathbf{X}) + \alpha_I^W \log \frac{\sum_i \bar{\lambda}_i \|\mathbf{p}_i - \frac{\sum_{i'} \bar{\lambda}_{i'} \mathbf{p}_{i'}}{\sum_{i'} \bar{\lambda}_{i'}} \|_2^2}{\sum_i \bar{\lambda}_i \|\mathbf{p}_i\|_2^2}. \qquad (19)$$

Intuitively, this is an explicit information bottleneck (IB) [43], where we maximize the (action) information about the classification label and meanwhile minimize the (scene) information about the non-action input frames.

4. Experiments

4.1. Datasets and Metrics

We evaluate SSAD and OSAD models on two standard benchmarks, THUMOS14 [13] and ActivityNet1.2 [2].

THUMOS14 consists of videos from 20 action classes. We follow the literature to train on validation set of 200

Table 1: Sanity check for the effect of random sampling in creating data splits for SSAD task. We randomly sample three data splits of ActivityNet1.2 in a fixed ratio of labeled/unlabeled data, and report the performance (mAP@AVG) of SSAD baselines on each split.

	split #1	split #2	split #3
MT	11.49	11.41	11.56
MixMatch	11.05	11.03	11.08
FixMatch	11.88	11.89	11.97

videos and evaluate on test set of 212 videos. This dataset has a fine level annotation of action. On average, each video lasts 3 minutes and contains 15.5 action clips. Duration of action instances varies from several seconds to minutes.

ActivityNet1.2 contains ~ 10k videos from 100 classes. Each video has an average of 1.5 action clips. Following the literature, we train our model on training set of 4819 videos and evaluate on validation set of 2383 videos.

Data Split. Both datasets are originally used in fully-supervised action detection, and we need to create labeled/unlabeled data splits for our SSAD task. We create the data splits by randomly sampling fully-labeled data in a specific ratio, and treat the rest as unlabeled data. The same process is also applied for creating three disjoint splits in OSAD task. We also have a sanity check to show that the random sampling will not affect the performance significantly as long as the ratio of each split is fixed (see Sec. 4.3).

Evaluation Metrics. Following the standard evaluation protocol, we report mean Average Precision (mAP) at different intersection over union (IoU) thresholds. We use $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ as the IoU thresholds for THUMOS14, and $\{0.5, 0.75, 0.95\}$ for ActivityNet1.2. We also report the average mAP over thresholds $[0.5 : 0.05 : 0.95]$.

4.2. Implementation Details

Input. We use RGB and optical flow as two separate input streams. RGB frames are sampled at 25fps for THUMOS14 and 3fps for ActivityNet1.2. Then optical flow is extracted from RGB frames using TV-L1 algorithm [30]. During training, we use sliding window to generate proposals of various durations. Then we sample 5 frames from each proposal by uniformly dividing the proposal into 5 segments and randomly sampling one frame from each segment. Features are extracted from each frame and averaged as the proposal feature. Following SSN [55], we additionally sample 2 frames in $[s_i - \frac{e_i - s_i}{2}, s_i]$ and 2 frames in $[e_i, e_i + \frac{e_i - s_i}{2}]$ to include the temporal context when predicting completeness or regression scores.

Backbone Model. We use BNInception [14] as the classification backbone architecture in SSN, and replace the feature pyramid (STPP) in the original SSN algorithm with a simple temporal pooling. We also adopt sliding window instead of TAG for proposal sampling because TAG needs

Table 2: Performance of SSAD models under 50% supervision on THUMOS14. We report results of supervised-only (Sup) model, as well as SSAD baselines with Mean Teacher (MT), MixMatch and FixMatch, with or without UFA. We also list the results under 100% supervision as reference.

Method	UFA	0.3	0.4	0.5	0.6	0.7
Sup (100%)	-	51.48	40.40	28.45	16.88	7.77
Sup	-	45.37	35.03	24.74	14.66	6.45
MT	-	43.74	34.62	24.62	14.90	6.87
	✓	45.47	35.52	25.43	15.35	7.08
MixMatch	-	44.79	35.80	24.79	14.76	6.60
	✓	**45.65**	**36.43**	**26.18**	**15.52**	**7.10**
FixMatch	-	42.94	33.19	23.58	14.10	6.61
	✓	42.99	34.65	24.68	14.72	6.61

Table 3: Performance of SSAD models under 10% supervision on ActivityNet1.2 (the same setting as THUMOS14).

Method	UFA	0.5	0.75	0.95	AVG
Sup (100%)	-	32.88	18.84	3.03	19.44
Sup	-	18.30	11.47	1.61	11.29
MT	-	18.71	11.55	1.55	11.49
	✓	19.11	11.75	1.63	11.74
MixMatch	-	17.84	11.27	1.61	11.05
	✓	18.73	11.91	1.82	11.84
FixMatch	-	18.63	12.03	1.68	11.88
	✓	**19.47**	**12.54**	**1.88**	**12.27**

Table 4: Comparison of different attention schemes. Here we report the performance of SSAD models with Mixmatch under 50% supervision on THUMOS14.

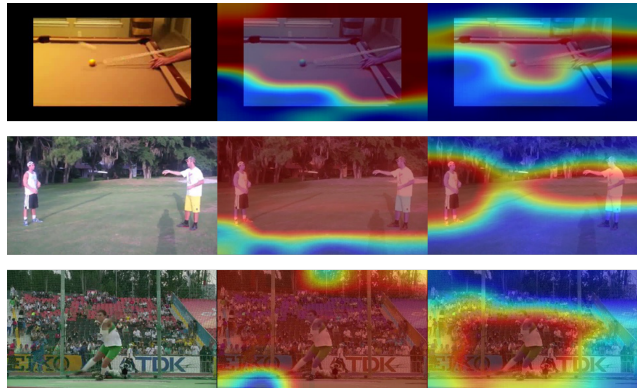|	0.3	0.4	0.5	0.6	0.7
w/o att	44.79	35.80	24.79	14.76	6.60
Gaussian	45.04	35.78	24.59	14.84	6.55
CIS	44.88	35.85	25.01	15.04	6.46
UFA (ours)	**45.65**	**36.43**	**26.18**	**15.52**	**7.10**

Figure 4: Visualization of the foreground attention from three example videos (One frame per video is shown). Columns from left to right: original frame, CIS attention, and our proposed UFA.

pretraining on unavailable labeled data.

Error Analysis. We consider three common types of errors: 1) **Action incompleteness**: missing parts of action; 2) **Misclassification**: incorrect action classification; 3) **Action-context confusion**: recognizing non-action frames as action. Error analysis results are shown in Fig. 1 where Mixmatch is used in both SSAD and OSAD models.

Hyperparameters. Please refer to the supplementary.

4.3. Semi-Supervised Action Detection

Before evaluating SSAD models, we first have a sanity check to show that randomly sampling in creating data splits will not greatly affect the result. We randomly sample 10% labeled data from ActivityNet1.2 for three times and test the results of SSAD baselines with three SSL algorithms. As shown in Table 1, the fluctuation in mAP@AVG is less than 0.1, so we experiment in a fixed random split.

Table 2 shows the performance of SSAD models on THUMOS14. Since each class only has 10 videos in THUMOS14, it will be more like a "few-shot" setting (1 video per class) if we choose the split of 10% labeled data which is normally adopted in SSL community. Thus, we test on the data split of 50% / 50% (labeled / unlabeled). We also report the result with 100% labeled data for reference. As we can see, the SSAD baselines barely improve the performance over supervised-only model due to the action incompleteness issue (Fig. 1). The FixMatch and MT methods even

slightly degrade the precision. When adding the UFA module, we improve the SSAD baselines by a large margin and fill in half the result gap between 50% supervised and 100% supervised models with no extra labels.

SSAD results on ActivityNet1.2 are shown in Table 3. We test on the data split of 10% / 90% (labeled / unlabeled) as an usual practice in SSL. Similar to THUMOS14, we observe no improvement in Mean Teacher and MixMatch baselines, and an improvement of 0.5% in FixMatch baseline. Our proposed UFA brings an accuracy boost of 0.5%-1% on average for all the three SSAD baselines.

We further ablate the proposed UFA module, and compare it with a simple Gaussian attention and a state-of-the-art unsupervised foreground detection method called CIS [52]. The Gaussian attention is a simple baseline which utilizes a Gaussian distribution centered in the image as a fixed attention map. CIS learns to detect foreground based on the assumption of independence between foreground and background motion in the same frame, which neglects the factor of camera motion. As shown in Table 4, Gaussian and CIS have slight or even no improvement over the baseline without attention. For an intuitive understanding, we also visualize the attention map from both CIS and our UFA in Fig 4. As expected, our UFA module exhibits more clear foreground attention while suppressing the background, and further helps recognize complete action.

Table 5: OSAD results on THUMOS14. We evaluate on two data splits, 1:2:7 and 1:4:5 (fully-/weakly-/un-labeled). We report performances of supervised-only, SSAD and OSAD/OSAD-IB. We also report the 100% fully-supervised performance for reference.

Data split	Supervision	mAP@0.5
100% / 0% / 0%	Sup. only	28.45
10% / 20% / 70%	Sup. only	8.08
	SSAD	8.96
	OSAD	11.37
	OSAD-IB	**12.97**
10% / 40% / 50%	Sup. only	8.08
	SSAD	8.96
	OSAD	14.26
	OSAD-IB	**15.33**

4.4. Omni-Supervised Action Detection

We evaluate the OSAD models on two sets of data splits, 10%/20%/70% and 10%/40%/50% (fully-/weakly-/un-labeled), on both datasets. In all experiments, we unify the experimental setting of using Mixmatch on unlabeled data with UFA. Table 5 shows the OSAD results on THUMOS14. With weakly-labeled data, the OSAD baseline brings a significant improvement over the SSAD model. The final model OSAD-IB further boosts the accuracy.

The OSAD results on ActivityNet1.2 are shown in Table 6. Notably, introducing weakly-labeled data in the OSAD baseline only has a small impact compared to SSAD, due to the action-context confusion. With the proposed IB added to our full OSAD-IB model, the action-context confusion issue is reduced and a better performance is obtained.

4.5. Multi-Level Supervision is More Efficient

Practically, when confronting a specific application scenario, as a first step we need to collect and annotate some training data. Suppose we have a fixed annotation budget which allows annotating limited amount of data, then designing the best annotation policy to maximize the detection performance is particularly useful. To be more specific, we need to decide whether to annotate more less-expensive weak data, or less more-expensive data with full supervision, or adopt a mixed strategy. We simulate this scenario on THUMOS14 to explore different annotation strategies and show the benefit of multi-level supervision.

As estimated from our user studies, the ratio of full and weak annotation cost on THUMOS14 is about $8 : 1$ (see supplementary). If we assume the weak annotation costs 1 time unit for each video, then the cost of fully annotating all 200 videos in THUMOS14 would be 1600 units. We assume only a 20% budget is available, *i.e.*, $1600 \times 0.2 = 320$ units. We test three annotation policies: 1) **Full**: use all the budget on full supervision; 2) **Weak**: use all the budget on weak supervision, then on full supervision if there is any left; 3) **Mixed**: trade-off between Full and Weak.

Table 6: OSAD results on ActivityNet1.2.

Data split	Method	mAP@AVG
100% / 0% / 0%	Sup. only	19.44
10% / 20% / 70%	Sup. only	11.29
	SSAD	11.84
	OSAD	11.78
	OSAD-IB	**12.30**
10% / 40% / 50%	Sup. only	11.29
	SSAD	11.84
	OSAD	12.44
	OSAD-IB	**13.25**

Table 7: Different annotation policies under a fixed budget on THUMOS14. All the models use Mixmatch with UFA and IB.

Policy	$|\mathcal{S}|$ (%)	$|\mathcal{W}|$ (%)	$|\mathcal{U}|$ (%)	0.3	0.5	0.7
Full	20%	0%	80%	34.02	16.06	**3.92**
Mixed	15%	40%	45%	**37.08**	**16.71**	3.68
	10%	80%	20%	34.39	15.26	3.76
Weak	8%	92%	0%	33.80	15.37	3.32

From Table 7, we observe that the Weak policy gives the worst result, which means full supervision is very important. However, spending all the budget on full supervision is also suboptimal in terms of results, and the best strategy is mixing full and weak supervision. The performance reaches a peak at $|\mathcal{S}| : |\mathcal{W}| = 15\% : 40\%$, suggesting that multi-level supervision is more efficient than the common strategy with only full or weak supervision.

5. Conclusion

In this work we explore multi-level supervision in temporal action detection. We first introduce the semi-supervised action detection (SSAD) task to learn with both fully-labeled and unlabeled videos. We build SSAD baselines by combining the fully-supervised action detection backbone with state-of-the-art semi-supervised learning algorithms. An unsupervised foreground attention (UFA) module is proposed to alleviate the action incompleteness issue in SSAD baselines by extracting object-centric features. Then we study the task of omni-supervised action detection (OSAD) where weakly-labeled videos are further incorporated. To resolve the action-context confusion in OSAD baselines, we design an information bottleneck (IB) to filter out scene information while keeping the action information, so that action is better distinguished from context frames. We conduct extensive experiments on SSAD and OSAD baselines, as well as the proposed UFA and IB methods, showing their effectiveness. We further show the advantage of multi-level supervision over single supervision under a realistic scenario with a fixed annotation budget.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019. 1, 3, 4

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. 6

[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. 3

[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*, pages 1130–1139, 2018. 1, 2

[5] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017. 2, 3

[6] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A flexible model for training action localization with varying levels of supervision. *arXiv preprint arXiv:1806.11328*, 2018. 3

[7] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *Advances in Neural Information Processing Systems*, pages 853–865, 2019. 3

[8] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013. 2

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 2

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 2

[11] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[13] 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 (CVIU)*, 155:1–23, 2017. 6

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3, 6

[15] Jingwei Ji, Kaidi Cao, and Juan Carlos Niebles. Learning temporal action proposals with fewer labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7073–7082, 2019. 3

[16] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 2

[17] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013. 2

[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. 1, 2

[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*, pages 1298–1307, 2019. 2, 3, 6

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016. 2

[21] Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. *arXiv preprint arXiv:2004.00163*, 2020. 2

[22] 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 *European conference on computer vision*, pages 420–437. Springer, 2020. 3

[23] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2020. 2, 3

[24] Tomas McCandless and Kristen Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, volume 2, page 3. Citeseer, 2013. 2, 3

[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*, pages 8679–8687, 2019. 3

[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*, pages 6752–6761, 2018. 2, 3

[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*, pages 5502–5511, 2019. 2, 3

[28] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018. 1, 2

[29] 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. 2, 3

[30] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line (IPOL)*, 2013:137–150, 2013. 6

[31] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5533–5541, 2017. 3

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*, pages 91–99, 2015. 2

[34] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G Schwing, and Jan Kautz. Ufo: A unified framework towards omni-supervised object detection. In *European Conference on Computer Vision*, pages 288–313. Springer, 2020. 3

[35] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1019, 2020. 2, 3, 6

[36] 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*, pages 5734–5743, 2017. 1, 2

[37] 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. 3

[38] 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. 2

[39] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 3

[40] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 1, 3

[41] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 715–731, 2018. 2, 3

[42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 1, 2, 3

[43] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 6

[44] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3386–3394, 2017. 2, 3

[45] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4481–4490, 2017. 2, 3

[46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015. 3

[47] 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. 3

[48] 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 *European conference on computer vision*, pages 20–36. Springer, 2016. 3

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018. 2, 3

[50] 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*, pages 5783–5792, 2017. 1, 2

[51] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 2

[52] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019. 2, 3, 5, 7

[53] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W. Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[54] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7094–7103, 2019. 2

[55] 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*, pages 2914–2923, 2017. 1, 2, 3, 4, 6

[56] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 3