Weakly Supervised Temporal Action Localization Through Learning Explicit Subspaces for Action and Context

Ziyi Liu[1], Le Wang[1*], Wei Tang[2], Junsong Yuan[3], Nanning Zheng[1], Gang Hua[4]
[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]University of Illinois at Chicago
[3]The State University of New York at Buffalo
[4]Wormpex AI Research
liuziyi@stu.xjtu.edu.cn, {lewang, nnzheng}@mail.xjtu.edu.cn,
tangw@uic.edu, jsyuan@buffalo.edu, ganghua@gmail.com

Abstract

Weakly-supervised Temporal Action Localization (WS-TAL) methods learn to localize temporal starts and ends of action instances in a video under only video-level supervision. Existing WS-TAL methods rely on deep features learned for action recognition. However, due to the mismatch between classification and localization, these features cannot distinguish the frequently co-occurring contextual background, *i.e.*, the context, and the actual action instances. We term this challenge action-context confusion, and it will adversely affect the action localization accuracy. To address this challenge, we introduce a framework that learns two feature subspaces respectively for actions and their context. By explicitly accounting for action visual elements, the action instances can be localized more precisely without the distraction from the context. To facilitate the learning of these two feature subspaces with only video-level categorical labels, we leverage the predictions from both spatial and temporal streams for snippets grouping. In addition, an unsupervised learning task is introduced to make the proposed module focus on mining temporal information. The proposed approach outperforms state-of-the-art WS-TAL methods on three benchmarks, *i.e.*, THUMOS14, ActivityNet v1.2 and v1.3 datasets.

1 Introduction

Temporal action localization (TAL) means to localize temporal starts and ends of action instances in an untrimmed video. It is a fundamental task in video understanding and has numerous applications such as intelligent surveillance, video retrieval and video summarization (Dan, Verbeek, and Schmid 2013; Asadiaghbolaghi et al. 2017; Kang and Wildes 2016). Since untrimmed videos can contain irrelevant content and multiple action instances, training fully supervised TAL methods (Chao et al. 2018; Lin et al. 2018, 2019; Liu et al. 2019a; Xu et al. 2020a) requires temporal boundary annotations of action instances. However, obtaining such annotations is time-consuming, which drives the research of weakly-supervised temporal action localization (WS-TAL) methods.

WS-TAL learns to localize action instances with only video-level categorical labels provided during training. However, the mismatch between this video-level classification supervision and the temporal localization task makes it difficult to distinguish the actual action instances from their frequently co-occurring contextual background (termed *context*). For example, the action baseball pitch commonly co-occurs with the context baseball field both temporally and spatially. We term this challenge *action-context confusion*.

Most existing methods train a video-level classifier and apply it snippet by snippet to achieve temporal action localization, as shown in Figure 1. Unfortunately, this pipeline can only exclude the irrelevant background but not the relevant context from the localization results. While the former is class-agnostic, the latter is action-specific: the context of an action seldom co-occurs with other actions. Therefore, irrelevant background is useless to classification, but context can provide important classification cues. For example, recognizing the baseball field will prevent the video from being misclassified as diving. Therefore, features trained on video-level categorical labels will look for contextual cues, and the entire learned feature space will represent action and context jointly. Consequently, snippets with only context are frequently mistaken as part of the action instance.

The goal of this paper is to address the challenge of action-context confusion in WS-TAL. Our idea is to learn a feature space which can be explicitly split into two (feature) subspaces, *i.e.*, the action subspace and the context subspace, as shown in Figure 1. After representing action visual elements in a feature subspace separated from context visual elements, the target action instances can be localized considering only action elements represented in the action subspace. Therefore, the distraction from the context is alleviated, in spite of its temporal and spatial co-occurrence with action.

However, learning explicit action and context subspaces with insufficient supervision is nontrivial. The key challenge is how to collect snippets with action/only context/only irrelevant background (*i.e.*, red dots/black dots/circles in Figure 1). We take advantage of the two-stream video representation, *i.e.*, spatial (RGB) and temporal (optical flow) streams for snippets grouping. Specifically, we con-

*Corresponding author.

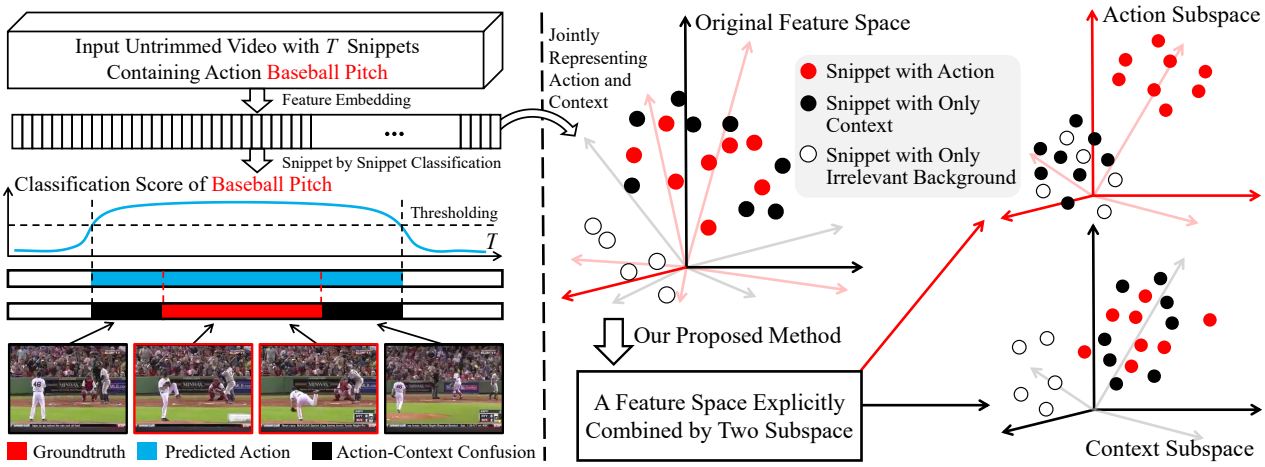

Figure 1: Illustration of a common WS-TAL pipeline with action-context confusion (left) and the main idea of our method (right). The original feature space jointly represents action and context visual elements, resulting in confusion of actions (red dots) and context (black dots). The proposed method learns an action subspace and a context subspace. By representing action visual elements in a separate feature subspace, we can alleviate the distraction from the context during temporal localization. In the action subspace, we require the black dots to be similar to snippets with only irrelevant background (circles) and distinct from red dots. While in the context subspace, the black dots are similar to red dots due to the spatial co-occurrence between action and context. We require both the black and red dots are distinct from circles in the context subspace.

sider snippets receiving consistent positive/negative predictions from both streams as snippets with action/only irrelevant background; snippets receiving inconsistent predictions from two streams are regarded as snippets with only context. Intuitively, snippets containing typical static scenes (such as a baseball field) provide useful appearance features but lack meaningful motion. Thus, they are considered positive by the spatial stream but negative by the temporal stream. Similarly, snippets containing typical non-action motion (such as the splash after diving or camera movement) are considered positive by the temporal stream but negative by the spatial stream. Besides, to further remedy the lack of supervision, we introduce an additional unsupervised training task. Together with a new temporal residual module, it makes our method focus on mining useful temporal information.

In summary, our contributions are as follows. (1) We propose to address the challenge of action-context confusion for WS-TAL by learning two feature subspaces, *i.e.*, the action subspace and the context subspace. It helps exclude the contextual background from the localization results. (2) We introduce a temporal residual module and an unsupervised training task to make our method focus on mining useful temporal information. (3) Our method achieves state-of-the-art performance on three benchmarks.

2 Related Work

TAL with Full Supervision TAL with full supervision requires temporal boundary annotations of the target action class during training. Similar to 2D object detection, the TAL task can be formulated as 1D object detection. Following its success in 2D object detection (Girshick 2015; Ren et al. 2017), the two-stage pipeline is leveraged by fully-supervised TAL methods (Escorcia et al. 2016; Shou, Wang, and Chang 2016; Buch et al. 2017; Gao et al. 2017; Xu, Das, and Saenko 2017; Zhao et al. 2017; Chao et al. 2018; Lin et al. 2018). Recently, instead of viewing temporal action proposals individually, the dependencies among them are considered. BMN (Lin et al. 2019) introduces the boundary-matching mechanism to capture the temporal context of each proposal. Graph Convolutional Networks (GCN) is a popular tool to capture the proposal-proposal relations, as proposed in P-GCN (Zeng et al. 2019), G-TAD (Xu et al. 2020b) and AGCN (Li et al. 2020).

TAL with Weak Supervision WS-TAL methods focus on achieving TAL with only video-level categorical labels. Without temporal boundary annotations for explicit supervision, the attention mechanism is widely used for distinguishing action and action-agnostic background. UntrimmedNet (Wang et al. 2017) formulates the attention mechanism as a soft selection module to localize target action. STPN (Nguyen et al. 2018) proposes a sparsity loss to improve UntrimmedNet and leverage multi-layer structure for attention learning. W-TALC (Paul, Roy, and Roy-Chowdhury 2018) proposes a co-activity loss to enforce the feature similarity among localized instances. AutoLoc (Shou et al. 2018) designs an "outer-inner-contrastive loss" for proposal evaluation, to facilitate the temporal boundary regression. CleanNet (Liu et al. 2019b) designs a "contrast score" by leveraging temporal contrast in SCPs to achieve end-to-end training of localization. BM (Nguyen, Ramanan, and Fowlkes 2019) achieves better TAL performance via background modelling and other unsupervised losses to guide the attention. CMSC (Liu, Jiang, and Wang 2019) exploits the action-context confusion challenge by regarding frames with low optical flow intensity as background.

However, the existing WS-TAL methods did not realize the learned feature represents action and context visual elements jointly, resulting the action-context confusion when temporally localizing the target action instances. By learning explicit subspaces for action and context, the proposed method can better avoid the distraction from the context, in spite of its temporal and spatial co-occurrence with action.

3 Proposed Method

The overview of the proposed architecture is shown in the left part of Figure 2. Identical architectures are applied for both spatial (RGB) and temporal (optical flow) streams. For notation simplicity, we omit the stream superscript without causing confusion in Section 3.1-3.4.

3.1 Feature Embedding

Given an untrimmed video, we first divide it into T non-overlapping snippets, which are the inputs of the feature embedding module. The outputs are the corresponding features of T snippets, denoted as $\mathbf{F}_o \in \mathbb{R}^{D_o \times T}$. $\mathbf{F}_o(t) \in \mathbb{R}^{D_o}$ is the feature of the t-th snippet s_t. We term the feature space of $\mathbf{F}_o$ "the original feature space", as shown in Figure 1.

It is worth noting that the original feature space of snippet-level features $\mathbf{F}_o$ jointly represents the visual elements of action and context, because they frequently co-occur both temporally and spatially and only video-level category labels are available during training. The Subspace module will address this issue as detailed in Section 3.3.

3.2 Base Module

The Base module performs video-level classification with the attention mechanism. Similar pipelines have been explored in (Nguyen et al. 2018; Nguyen, Ramanan, and Fowlkes 2019). This part is not included in our contribution.

After obtaining $\mathbf{F}_o$, the attention mechanism leverages $\mathbf{F}_o$ to obtain snippet-level predictions (SAPs):

$$\mathbf{a}(t) = \mathcal{M}_{\text{att}}(\mathbf{F}_o(t), \phi_{\text{att}}), \ \ \mathbf{a} \in \mathbb{R}^{1\times T}, \tag{1}$$

where $\mathcal{M}_{\text{att}}$ is a combination of a fully connected (FC) layer and a sigmoid function. ϕ_{att} is the corresponding learnable parameters. Afterwards, the video-level foreground and background features are fused by temporal weighted average pooling following (Nguyen, Ramanan, and Fowlkes 2019):

$$\mathbf{f}_{\text{fg}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{a}(t)\mathbf{F}_o(t), \ \mathbf{f}_{\text{fg}} \in \mathbb{R}^{D_o}, \tag{2}$$

$$\mathbf{f}_{\text{bg}} = \frac{1}{T}\sum_{t=1}^{T} (1-\mathbf{a}(t))\mathbf{F}_o(t), \ \mathbf{f}_{\text{bg}} \in \mathbb{R}^{D_o}. \tag{3}$$

After obtaining video-level features, video-level predictions are obtained as

$$\mathbf{p}_{\text{fg}} = \mathcal{M}_{\text{cls}}(\mathbf{f}_{\text{fg}}, \phi_{\text{cls}}), \ \mathbf{p}_{\text{fg}} \in \mathbb{R}^{(N+1)}, \tag{4}$$

$$\mathbf{p}_{\text{bg}} = \mathcal{M}_{\text{cls}}(\mathbf{f}_{\text{bg}}, \phi_{\text{cls}}), \ \mathbf{p}_{\text{bg}} \in \mathbb{R}^{(N+1)}, \tag{5}$$

where $\mathcal{M}_{\text{cls}}$ is implemented as an FC layer to get classification results. $(N+1)$ indicates the total number of action classes including the background class.

Applying $\mathcal{M}_{\text{cls}}$ to features of each snippet $\mathbf{F}_o(t)$, the snippet-level classification predictions are obtained as

$$\mathbf{P}_o(t) = \mathcal{M}_{\text{cls}}(\mathbf{F}_o(t), \phi_{\text{cls}}), \ \ \mathbf{P}_o \in \mathbb{R}^{(N+1)\times T}. \tag{6}$$

3.3 Subspace Module

The goal of the Subspace module is to transform the original feature space to a feature space explicitly combined by two subspaces, *i.e.*, the action subspace and the context subspace. The architecture is shown in the right part of Figure 2. First, we transform the feature of the t-th snippet $\mathbf{F}_o(t)$ as

$$\mathbf{F}(t) = \mathcal{M}_{\mathcal{T}}(\mathbf{F}_o(t), \phi_{\mathcal{T}}), \ \mathbf{F} \in \mathbb{R}^{2D\times T}, \tag{7}$$

where $\mathcal{M}_{\mathcal{T}}$ is implemented as an FC layer with a ReLU activation and $\phi_{\mathcal{T}}$ denotes its trainable parameters. The combined feature space is $2D$ dimensional, with D dimensions for each subspace. Such a transformation is snippet independent. To consider snippet contextual information, a temporal residual module (T-ResM) is leveraged to enhance the learned features:

$$\mathbf{F}_r = \mathbf{F} + \mathcal{M}_{\text{res}}(\mathbf{F}, \phi_{\text{res}}), \ \mathbf{F}_r \in \mathbb{R}^{2D\times T}, \tag{8}$$

where $\mathcal{M}_{\text{res}}$ is implemented using two stacked temporal 1d-convolutional layers, ϕ_{res} represents the learnable parameters. Note that the T-ResM is optional. With only video-level categorical labels available during training, additional layers such as $\mathcal{M}_{\text{res}}$ will adversely affect the training process due to the lack of supervision. Therefore, an unsupervised training task is introduced to facilitate the training of $\mathcal{M}_{\text{res}}$, which will be detailed in Section 3.4.

After obtaining features in the combined feature space (*i.e.*, $\mathbf{F}_r$), features in the action subspace $\mathbf{F}_a \in \mathbb{R}^{D\times T}$ and features in the context subspace $\mathbf{F}_c \in \mathbb{R}^{D\times T}$ are obtained by directly splitting $2D$ channels into two parts, each with D channels, *i.e.*,

$$\mathbf{F}_r = [\mathbf{F}_a, \mathbf{F}_c], \tag{9}$$

where $[\cdot]$ indicates vector concatenation.

Subsequently, for the t-th snippet, we can obtain its predictions from the action subspace ($\mathbf{P}_a(t) \in \mathbb{R}^{(N+1)}$) and the context subspace ($\mathbf{P}_c(t) \in \mathbb{R}^{(N+1)}$) as

$$\mathbf{P}_a(t) = \mathcal{M}_a(\mathbf{F}_a(t), \phi_a), \ \mathbf{P}_c(t) = \mathcal{M}_c(\mathbf{F}_c(t), \phi_c), \tag{10}$$

where $\mathcal{M}_a$ and $\mathcal{M}_c$ represent the classification layers for action and context subspaces with trainable parameters ϕ_a and ϕ_c, respectively.

Summary of the proposed architecture. For Base module, in the training stage, its outputs are (1) $\mathbf{p}_{\text{fg}} \in \mathbb{R}^{(N+1)}$ and $\mathbf{p}_{\text{bg}} \in \mathbb{R}^{(N+1)}$ for Base module training; (2) $\mathbf{a} \in \mathbb{R}^{1\times T}$ for Subspace module training. In the testing stage, the outputs of Base module are (1) $\mathbf{a}$ for temporal action proposal (TAP) generation; (2) $\mathbf{P}_o \in \mathbb{R}^{(N+1)\times T}$ for TAP evaluation.

For Subspace module, in the training stage, its outputs are (1) $\mathbf{P}_a \in \mathbb{R}^{(N+1)\times T}$ and $\mathbf{P}_c \in \mathbb{R}^{(N+1)\times T}$ for the training of Subspace module; (2) $\mathbf{F}_r \in \mathbb{R}^{2D\times T}$ for the training of T-ResM. In the testing stage, the output is $\mathbf{P}_a$ for both TAP generation and evaluation. The context subspace is only used for facilitating the training process and not considered for performing TAL.

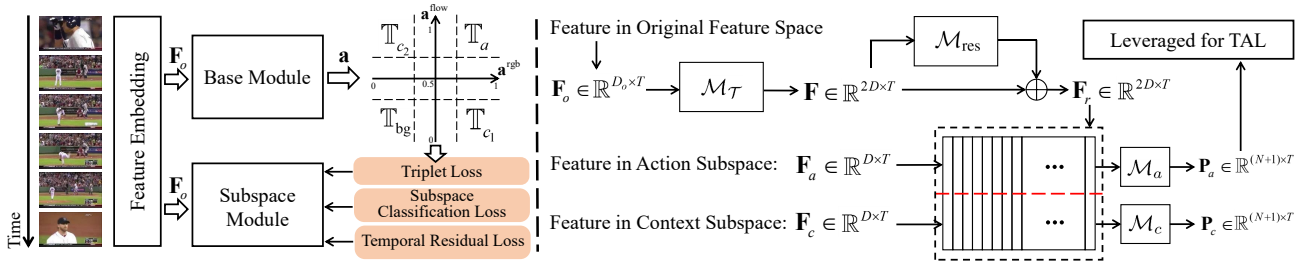

Figure 2: An overview of our method (left) and the detailed architecture of the proposed Subspace module (right). Left: our proposed method is mainly composed of three modules, *i.e.*, feature embedding, Base module and Subspace module. Leveraging the Base module, we divide snippets into four sets, *i.e.*, $\mathbb{T}_a, \mathbb{T}_{c_1}, \mathbb{T}_{c_2}$ and $\mathbb{T}_{\text{bg}}$ in Eq. (11-14). Right: Using the Subspace module, the snippet-level features in action and context subspaces are obtained. Only the output of action subspace is used for localization to alleviate the distraction from context.

The above architecture is applied for both spatial and temporal streams. Adding the superscript "rgb" or "flow" to notations above, we can obtain the notations for the spatial or temporal stream. This also applies the training process below.

3.4 Training Process for Classification

This section introduces how to learn explicit action and context subspaces with only video-level labels. The training of the Base module has been well explored and is not our main contribution. Please refer to our supplementary material for details.

An Overview of the Training of Subspace Module Our training objective is to ensure the obtained action/context subspace and the feature $\mathbf{F}_r \in \mathbb{R}^{2D\times T}$ have three properties. The **first** property is about the similarities among features in different subspaces, guided by the triplet loss L_t. The **second** property is about the classification predictions of features in different subspaces, guided by the subspace classification loss L_s. The **third** property is that $\mathbf{F}_r \in \mathbb{R}^{2D\times T}$ should contain temporal information, guided by an optional temporal residual loss L_r. The three properties and their corresponding losses are detailed below.

Preparation before Loss Calculation The sets of snippets with action/only context/only irrelevant background (denoted as $\mathbb{S}_a/\mathbb{S}_c/\mathbb{S}_{\text{bg}}$) are required for calculation of losses. Specifically, we first collect four snippet index sets as

$$\mathbb{T}_a = \{t \mid \mathbf{a}^{\text{rgb}}(t) > \theta_h \;\&\; \mathbf{a}^{\text{flow}}(t) > \theta_h\}, \tag{11}$$

$$\mathbb{T}_{c_1} = \{t \mid \mathbf{a}^{\text{rgb}}(t) > \theta_h \;\&\; \mathbf{a}^{\text{flow}}(t) < \theta_l\}, \tag{12}$$

$$\mathbb{T}_{c_2} = \{t \mid \mathbf{a}^{\text{rgb}}(t) < \theta_l \;\&\; \mathbf{a}^{\text{flow}}(t) > \theta_h\}, \tag{13}$$

$$\mathbb{T}_{\text{bg}} = \{t \mid \mathbf{a}^{\text{rgb}}(t) < \theta_l \;\&\; \mathbf{a}^{\text{flow}}(t) < \theta_l\}, \tag{14}$$

where θ_h and θ_l are the high and low thresholds, defined as

$$\theta_h = 0.5 + \alpha,\ \theta_l = 0.5 - \alpha, \tag{15}$$

where α is the hyper-parameter controlling the gap between θ_h and θ_l. Subsequently, $\mathbb{S}_a$ (or $\mathbb{S}_{\text{bg}}$) is obtained by collecting snippets with consistent positive (or negative) predictions from both streams, *i.e.*, $\mathbb{S}_a = \{s_t | t \in \mathbb{T}_a\}$ (or $\mathbb{S}_{\text{bg}} = \{s_t | t \in \mathbb{T}_{\text{bg}}\}$). Similarly, $\mathbb{S}_c$ is obtained by collecting snippets with inconsistent predictions from two streams as $\mathbb{S}_c = \{s_t | t \in \mathbb{T}_{c_1} \cup \mathbb{T}_{c_2}\}$.

First Property and Triplet Loss The first property: In the action subspace, the representation of $s_t \in \mathbb{S}_c$ should be similar with that of $s_t \in \mathbb{S}_{\text{bg}}$, because both sets of snippets contain no action. Meanwhile, the representation of $s_t \in \mathbb{S}_c$ should be distinct from that of $s_t \in \mathbb{S}_a$. On the contrary, in the context subspace, the representation of $s_t \in \mathbb{S}_c$ should be distinct from that of $s_t \in \mathbb{S}_{\text{bg}}$. Meanwhile, the representation of $s_t \in \mathbb{S}_c$ should be similar to that of $s_t \in \mathbb{S}_a$ because of the spatial co-occurrence between action and context. This property can be naturally guided by the triplet loss.

In the action subspace, we obtain three video-level features representing action, context and irrelevant background snippets as

$$\mathcal{A}_a = \frac{1}{|\mathbb{T}_a|}\sum_{t\in\mathbb{T}_a}\mathbf{F}_a(t), \tag{16}$$

$$\mathcal{A}_c = \frac{1}{|\mathbb{T}_c|}\sum_{t\in\mathbb{T}_c}\mathbf{F}_a(t), \tag{17}$$

$$\mathcal{A}_{\text{bg}} = \frac{1}{|\mathbb{T}_{\text{bg}}|}\sum_{t\in\mathbb{T}_{\text{bg}}}\mathbf{F}_a(t), \tag{18}$$

where $|\cdot|$ denotes the number of elements. Similarly, in the context subspace, we obtain video-level features representing action, context and irrelevant background snippets as

$$\mathcal{C}_a = \frac{1}{|\mathbb{T}_a|}\sum_{t\in\mathbb{T}_a}\mathbf{F}_c(t), \tag{19}$$

$$\mathcal{C}_c = \frac{1}{|\mathbb{T}_c|}\sum_{t\in\mathbb{T}_c}\mathbf{F}_c(t), \tag{20}$$

$$\mathcal{C}_{\text{bg}} = \frac{1}{|\mathbb{T}_{\text{bg}}|}\sum_{t\in\mathbb{T}_{\text{bg}}}\mathbf{F}_c(t). \tag{21}$$

Afterwards, we leverage the triplet loss as

$$L_t = \max(\bar{d}(\mathcal{A}_c, \mathcal{A}_{\text{bg}}), \bar{d}(\mathcal{A}_c, \mathcal{A}_a) + m, 0) + \max(\bar{d}(\mathcal{C}_c, \mathcal{C}_a) - \bar{d}(\mathcal{C}_c, \mathcal{C}_{\text{bg}}) + m, 0), \tag{22}$$

where $\bar{d}(\mathbf{p}, \mathbf{q})$ is the Euclidean distance between the ℓ_2 normalized $\mathbf{p}$ and $\mathbf{q}$. m is the margin set as 1.

Second Property and Subspace Classification Loss The second property: In the action subspace, $\forall s_t \in \mathbb{S}_a$ should be predicted as having target action class (Eq. (23)). While $\forall s_t \in \mathbb{S}_c \cup \mathbb{S}_{\text{bg}}$ should be predicted as the background class (Eq. (24)). In the context subspace, $\forall s_t \in \mathbb{S}_c$ should be classified as the target action class (Eq. (25)), while $\forall s_t \in \mathbb{S}_{\text{bg}}$ should be predicted as the background class (Eq. (26)). No explicit constraint is imposed on the predictions of $s_t \in \mathbb{S}_a$ in the context subspace.

To obtain the subspace classification loss L_s, the snippet-level classification labels of predictions from action and context subspaces (*i.e.*, $\mathbf{P}_a \in \mathbb{R}^{(N+1)\times T}$ and $\mathbf{P}_c \in \mathbb{R}^{(N+1)\times T}$) are required. Following the second property, we assign the labels to $\mathbf{P}_a$ and $\mathbf{P}_c$ as

$$\forall t \in \mathbb{T}_a,\ \mathbf{P}_a(t)|_n \leftarrow 1,\ \mathbf{P}_a(t)|_0 \leftarrow 0, \quad (23)$$

$$\forall t \in \mathbb{T}_c \cup \mathbb{T}_{\text{bg}},\ \mathbf{P}_a(t)|_n \leftarrow 0,\ \mathbf{P}_a(t)|_0 \leftarrow 1, \quad (24)$$

$$\forall t \in \mathbb{T}_c,\ \mathbf{P}_c(t)|_n \leftarrow 1,\ \mathbf{P}_c(t)|_0 \leftarrow 0, \quad (25)$$

$$\forall t \in \mathbb{T}_{\text{bg}},\ \mathbf{P}_c(t)|_n \leftarrow 0,\ \mathbf{P}_c(t)|_0 \leftarrow 1, \quad (26)$$

where the video is annotated as having the n-th action class and $|_n$ indicates the classification prediction of the n-th action. The 0-th class indicates the background class.

After assigning the predictions with binary expectations as Eq. (23-26), we can train the Subspace module using a binary logistic regression loss L_s. Please refer to our supplementary material for detailed formula derivation of the L_s based on the predictions and binary expectations.

Third Property and Temporal Residual Loss The third property: $\mathbf{F}_r$ should contain temporal information as T-ResM is the only module that can provide cross-snippet information in the Subspace module. However, we observe that without additional constraints, temporal 1d-convolutional layers cannot enhance the temporal features. So we introduce an auxiliary training task to ensure that property, and the temporal residual loss L_r is the corresponding loss of this task.

Specifically, the auxiliary unsupervised task is "snippet-level four-class classification". Regarding snippet indexes belonging to different index sets as different classes, we collect four classes indicated by Eq. (11-14). Afterwards, we perform the four-class prediction as

$$\mathbf{P}_r(t) = \mathcal{M}_r(\mathbf{F}_r(t), \phi_r),\ \mathbf{P}_r \in \mathbb{R}^{4\times T}, \quad (27)$$

where $\mathcal{M}_r$ is a classification layer only used for this additional task. L_r is the corresponding cross-entropy loss:

$$L_r = -\frac{1}{|\mathbb{T}'|}\sum_{t\in\mathbb{T}'}\log(\mathbf{P}_r(t)|_n), \quad (28)$$

where $\mathbb{T}' = \{\mathbb{T}_a \cup \mathbb{T}_{c_1} \cup \mathbb{T}_{c_2} \cup \mathbb{T}_{\text{bg}}\}$, and $n = 0/1/2/3$ if $t \in \mathbb{T}_a/\mathbb{T}_{c_1}/\mathbb{T}_{c_2}/\mathbb{T}_{\text{bg}}$. This prediction involves cross stream information. To achieve this goal, $\mathbf{F}_r$ is required to account for snippet contextual information. Since the Subspace module of each stream is trained independently and T-ResM is the only module that can provide cross-snippet information, minimizing L_r can guide the T-ResM to focus on mining snippet contextual information.

Finally, the total loss for the training of the Subspace module is $L = L_t + L_s + L_r$.

3.5 Testing Process for TAL

The TAL results can be achieved using the outputs from the Base module or the Subspace module in the testing process. Firstly the outputs from two streams are fused by weighted average with a hyper-parameter β, *i.e.*,

$$\bar{\mathbf{a}} = \beta\mathbf{a}^{\text{rgb}} + (1-\beta)\mathbf{a}^{\text{flow}},\ \bar{\mathbf{a}} \in \mathbb{R}^{1\times T}, \quad (29)$$

$$\bar{\mathbf{P}}_o = \beta\mathbf{P}_o^{\text{rgb}} + (1-\beta)\mathbf{P}_o^{\text{flow}},\ \bar{\mathbf{P}}_o \in \mathbb{R}^{(N+1)\times T}, \quad (30)$$

$$\bar{\mathbf{P}}_a = \beta\mathbf{P}_a^{\text{rgb}} + (1-\beta)\mathbf{P}_a^{\text{flow}},\ \bar{\mathbf{P}}_a \in \mathbb{R}^{(N+1)\times T}. \quad (31)$$

To achieve TAL, there are two necessary steps, *i.e.*, temporal action proposal (TAP) generation and TAP evaluation. With only Base module, the TAP generation is achieved by thresholding $\bar{\mathbf{a}}$ using 0.5 (C_1 in Table 1). Given a TAP, its confidence score containing the n-th action is evaluated by Outer-Inner-Contrastive (Shou et al. 2018) as

$$s(t_s, t_e, n, \bar{\mathbf{P}}_o) = \text{mean}(\bar{\mathbf{P}}_o(t_s : t_e)|_n) - \text{mean}([\bar{\mathbf{P}}_o(t_s - \tau : t_s)|_n, \bar{\mathbf{P}}_o(t_e : t_e + \tau)|_n]), \quad (32)$$

where t_s and t_e are the starting and ending snippet indexes of the given TAP. $\bar{\mathbf{P}}_o(t_s : t_e)|_n$ indicates the fused prediction scores of the n-th action from the t_s-th snippet to the t_e-th snippet. $\tau = (t_e - t_s)/4$ denotes the inflation length and mean$(\cdot)$ is the mean function.

Instead of using Base module for TAL, our contribution is to leverage the output of the Subspace module (*i.e.*, $\mathbf{P}_a$) to achieve better TAL results, as evaluated in Table 4. Without the attention mechanism in Subspace module, the TAP generation is achieved by thresholding the sum of all non-background classes ($\sum_{n=1}^{N}\bar{\mathbf{P}}_a|_n$) using 0.5 ($C_2$ in Table 4). For TAP evaluation, we use $\bar{\mathbf{P}}_a$ to replace $\bar{\mathbf{P}}_o$ in Eq. (32) for better evaluation (C_3 in Table 4) because $\bar{\mathbf{P}}_a$ can better avoid the distraction from the context.

4 Experiments

4.1 Experimental Setting

Evaluation Datasets. THUMOS14. (Jiang et al. 2014) We use the subset of the THUMOS14 dataset with temporal boundary annotations provided. It includes 20 action categories. Following conventions, we train the proposed method using the validation set with 200 untrimmed videos and evaluate it on the test set with 213 untrimmed videos. On average, each video contains 15.4 action instances and 71.4% frames are non-action background. **ActivityNet.** (Fabian Caba Heilbron and Niebles 2015) The v1.2/v1.3 provides temporal boundary annotations for 100/200 activity classes with a total of $9,682/19,994$ videos. Since the temporal boundary annotations of the test set is not publicly available, we conduct experiments and report the performance on the validation set. The training set includes $4,819/10,024$ untrimmed videos and the validation set includes $2,383/4,926$ untrimmed videos. On average, each

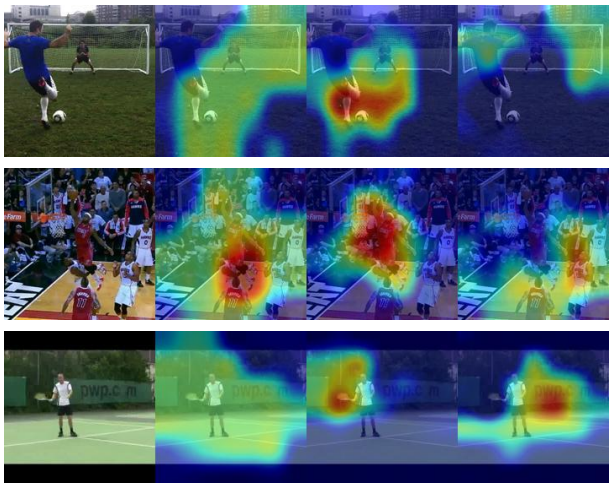

Image | The Original Feature Space | Action Subspace | Context Subspace

Figure 3: The spacial visualization of the original feature space, action subspace and context subspace. The original feature space jointly represents both action and context visual elements. Compared with the context subspace, the action subspace focuses on the action areas (soccer penalty, basketball dunk and tennis swing) instead of context areas (soccer field, basketball court and tennis court).

	Explanation
C_1	Thresholding $\bar{\mathbf{a}}$ with 0.5 for TAP generation.
C_2	Thresholding $\sum_{n=1}^{N} \bar{\mathbf{P}}_a\|_n$ with 0.5 for TAP generation.
C_3	Using $\bar{\mathbf{P}}_a$ to replace $\bar{\mathbf{P}}_o$ in Eq. (32) for TAP evaluation.
C_4	Using T-ResM to enhance $\mathbf{F}$ following Eq. (8).

Table 1: Four notations indicating different experiment settings used for ablation studies. C_1 is defined for comparison with Based module. C_2 and C_3 respectively represent the contribution of the Subspace module towards TAP generation and evaluation. C_4 is defined to solely evaluate the effect of T-ResM.

video contains 1.5/1.6 action instances and 34.6%/35.7% non-action background, which indicates a relatively lower noise ratio compared with THUMOS14.

Evaluation metric. Following conventions, we evaluate the TAL performance by mean average precision (mAP) values at different IoU thresholds. For the evaluation on THUMOS14, the IoU thresholds are $[0.1 : 0.1 : 0.9]$. For ActivityNet, the IoU thresholds are $[0.5 : 0.05 : 0.95]$.

4.2 Ablation Study

Spatial Visualization of Action Subspace. To validate the learned action subspace focuses on the action visual elements, we visualize the predictions in it spatially. To achieve the spatial visualization, we use the features before global average pooling and obtain $\bar{\mathbf{P}}_o$, $\bar{\mathbf{P}}_a$ and $\bar{\mathbf{P}}_c$ for every spatial location on the feature maps. Facilitated by snippets with only context (*i.e.*, $\mathbb{S}_c$) as negative training samples, $\bar{\mathbf{P}}_a$ can distinguish action from context. However, due to actions seldom appear without their context spatially, our method is lack of snippets with **only** action (*e.g.*, snippets with action soccer penalty but without a soccer field), which makes $\bar{\mathbf{P}}_c$ cannot directly distinguish context from action. To eliminate action elements in $\bar{\mathbf{P}}_c$, we use $\max(\bar{\mathbf{P}}_c - \bar{\mathbf{P}}_a, 0)$ for every spatial location to illustrate the context subspace. As shown in Figure 3, the outputs of different feature spaces are visualized by heat maps imposed on the original images.

α	β	mAP(%)@IoU 0.3	0.4	0.5	0.6	0.7	AVG
0#($\beta=0.4$)		38.4	30.5	21.5	14.5	7.3	22.4
0#($\beta=0.5$)		31.4	23.4	15.8	9.5	4.8	17.0
0#($\beta=0.6$)		27.2	20.3	13.6	7.6	4.0	14.5
0.1	0.4	50.6	**41.7**	29.7	20.2	10.4	30.5
	0.5	49.7	41.3	30.1	19.6	10.6	30.3
	0.6	49.9	41.0	29.9	19.7	10.1	30.1
0.2	0.4	**50.8**	**41.7**	29.6	20.1	10.7	**30.6**
	0.5	50.4	**41.7**	**30.3**	20.0	10.4	30.5
	0.6	48.8	40.3	29.0	19.8	9.8	29.5
0.3	0.4	50.2	41.4	29.9	20.0	10.6	30.4
	0.5	48.8	40.8	30.1	20.0	10.1	30.0
	0.6	49.2	39.5	28.3	18.6	9.3	29.0
0.4	0.4	49.7	41.6	30.0	**20.6**	10.5	30.5
	0.5	47.6	40.3	29.7	20.3	**10.9**	29.8
	0.6	47.9	38.5	27.5	17.6	9.7	28.3

Table 2: Sensitivity test of hyper-parameters α (in Eq. (15)) and β in Eq. (29-31) on THUMOS14 test set.

Effect of Each Component on TAL Task. To evaluate the contribution of the proposed Subspace module, we first define notations indicating different experiment settings as listed in Table 1. The effect of the proposed Subspace module on the TAL task is reflected in two aspects, *i.e.*, using $\bar{\mathbf{P}}_a$ to improve both TAP generation and TAP evaluation steps (C_2 and C_3 in Table 1). Comparing 1# (or 2#) with 0#, we can solely evaluate the contribution of C_3 (or C_2). Combining C_2 and C_3, the major improvement (48.8% in UNT and 78.5% in I3D) is achieved against the Base module. Facilitated by the T-ResM with the proposed unsupervised training task, the TAL performance can be further improved, as validated by the comparison between 3# and 4#. Compared with using I3D features, T-ResM plays a more important role when using UNT features, due to UNT features contain less temporal information compared with I3D features. Our proposed components (*i.e.*, C_2, C_3 and C_4) contribute to most of the performance gain (89.5% in UNT and 90.8% in I3D). Finally, with all the components, 5# achieves the best performance with both feature backbones.

Hyper-parameter Sensitivity. To explore the impact of the hyper-parameters, we adopt different settings of α and β, as summarized in Table 2. The Base module only depends on β. Compared with results from Base module, our method, strengthened by the proposed Subspace module, shows robustness towards all hyper-parameters.

Ablated Variants	Feature	C_1	C_2	C_3	C_4	mAP(%)@IoU 0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	AVG (0.3:0.7)	Gain
0#	UNT	✓				50.0	43.6	35.7	29.0	21.7	14.2	7.4	2.5	0.3	21.6	-
1#	UNT	✓		✓		51.3	45.4	37.3	30.4	22.5	15.0	7.8	2.7	0.3	22.6	$1.0\|_{21.1\%}$
2#	UNT		✓			50.9	45.1	38.3	30.0	22.9	15.4	7.7	2.4	0.3	22.8	$1.2\|_{26.2\%}$
3#	UNT		✓	✓		52.8	47.6	40.6	31.2	23.5	15.9	8.2	2.3	0.2	23.9	$2.3\|_{48.8\%}$
4#	UNT		✓	✓	✓	**54.8**	48.9	41.8	33.7	26.1	18.0	9.4	3.4	**0.5**	25.8	$4.2\|_{89.5\%}$
5#	UNT	✓	✓	✓	✓	54.7	**49.1**	**42.1**	**34.2**	**26.7**	**18.5**	**9.7**	**3.5**	**0.5**	**26.3**	$4.6\|_{100\%}$
0#	I3D	✓				48.9	44.7	38.4	30.5	21.5	14.5	7.3	2.6	0.3	22.4	-
1#	I3D	✓		✓		58.9	54.8	47.4	38.3	26.9	17.7	8.9	3.2	0.3	27.9	$5.5\|_{66.6\%}$
2#	I3D		✓			51.6	47.9	41.4	33.1	23.5	15.3	8.8	3.4	**0.5**	24.4	$2.0\|_{24.5\%}$
3#	I3D		✓	✓		60.2	56.4	49.2	40.1	27.6	17.7	9.6	3.2	**0.5**	28.8	$6.4\|_{78.5\%}$
4#	I3D		✓	✓	✓	**62.2**	**58.2**	50.7	40.6	28.3	19.2	10.4	4.0	**0.5**	29.8	$7.4\|_{90.8\%}$
5#	I3D	✓	✓	✓	✓	61.7	58.0	**50.8**	**41.7**	**29.6**	**20.1**	**10.7**	**4.3**	**0.5**	**30.6**	$8.2\|_{100\%}$

Table 4: Ablation studies of our method on the THUMOS14 dataset by using mAP under different IoU thresholds and the average mAP under the IoU thresholds from 0.3 to 0.7. Notations indicating experiment settings are defined in Table 1. UNT and I3D represent UntrimmedNet and I3D feature backbones, respectively. The gain of average mAP against the Base module (0#) of every variants are listed. Our proposed components (*i.e.*, C_2, C_3 and C_4) contribute most of the performance gain.

Method	v1.2 / v1.3	mAP(%)@IoU 0.5	0.75	0.95	Avg
AutoLoc (2018)	v1.2	27.3	15.1	3.3	16.0
TSM (2019)	v1.2	28.3	17.0	3.5	17.1
CleanNet(2019b)	v1.2	37.1	20.3	5.0	21.6
CMCS (2019)	v1.2	36.8	22.0	5.6	22.4
RPNet (2020)	v1.2	37.6	23.9	5.4	23.3
Ours	v1.2	**39.2**	**25.6**	**6.8**	**25.5**
TSM (2019)	v1.3	30.3	19.0	4.5	-
CMCS (2019)	v1.3	34.0	20.9	**5.7**	21.2
BM (2019)	v1.3	**36.4**	19.2	2.9	-
BaSNet (2020)	v1.3	34.5	22.5	4.9	22.2
Ours	v1.3	35.1	**23.7**	5.6	**23.2**

Table 3: TAL performance comparison on ActivityNet v1.2 and v1.3 validation set, in terms of average mAP at IoU thresholds $[0.5 : 0.05 : 0.95]$. All results are obtained using I3D features.

	Method	Feature	mAP@IoU 0.4	0.5	0.6	AVG
Full	SSN (2017)	UNT	41.0	29.8	19.6	30.6
	MGG (2019a)	I3D	46.8	37.4	29.5	37.8
Weak	AutoLoc (2018)	UNT	29.0	21.2	13.4	21.0
	CleanNet(2019b)	UNT	30.9	23.9	13.9	22.6
	RPNet (2020)	UNT	29.4	21.2	13.9	21.8
	Ours	UNT	**34.2**	**26.7**	**18.5**	**26.3**
Weak	MAAN (2019)	I3D	30.6	20.3	12.0	22.2
	CMCS(2019)	I3D	32.1	23.1	15.0	23.7
	BM (2019)	I3D	37.5	26.8	17.6	27.5
	ASSG (2019)	I3D	38.7	25.4	15.0	27.2
	BaSNet (2020)	I3D	36.0	27.0	18.6	27.3
	RPNet (2020)	I3D	37.2	27.9	16.7	27.6
	Ours	I3D	**41.7**	**29.6**	**20.1**	**30.6**

Table 5: Results on the THUMOS14 test set. We report average mAP values at IoU thresholds 0.3:0.1:0.7. Recent works in both full and weak supervision settings are reported. Our method outperforms the state-of-the-art methods on both backbone settings.

4.3 Comparisons with State-of-the-Art Methods

As presented in Table 5, our method significantly outperforms existing WS-TAL methods with both feature backbones on THUMOS14. As presented in Table 3, our method also achieves the best average mAP compared with previous WS-TAL methods on ActivityNet v1.2 and v1.3. However, the performance improvement is not as significant as that on THUMOS14, possibly due to the lower non-action frame rate in ActivityNet v1.2 and v1.3. With less distraction from non-action context, the improvement brought by our method will be less significant.

In summary, the effectiveness of our method is validated by extensive experiments on different feature backbones and benchmarks. Via spatial visualization, we validate that the original feature space indeed jointly represents both action and context visual elements. While in the learned action subspace, action visual elements are separated from context visual elements. Through detailed ablation studies, the contribution of the proposed Subspace module is evaluated, and the robustness towards hyper-parameters is validated. Finally, we have compared our method with state-of-the-art WS-TAL methods on three standard benchmarks, and significant improvement is observed.

5 Conclusion

We propose to address the action-context confusion challenge for WS-TAL, by learning explicit action and context subspaces. Leveraging the predictions from spatial and temporal streams for snippets grouping and introducing an auxiliary unsupervised training task, the two subspaces are learned and the distraction from the context is better avoided during temporal localization. Our method significantly outperforms existing WS-TAL methods on three standard datasets. Visualization results and detailed ablation studies further validate the contribution of the proposed method.

Acknowledgments

This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 61629301, 61773312, and 61976171, China Postdoctoral Science Foundation Grant 2019M653642, Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001, and Natural Science Foundation of Shaanxi Grant 2020JQ-069.

References

Asadiaghbolaghi, M.; Clapes, A.; Bellantonio, M.; Escalante, H. J.; Poncelopez, V.; Baro, X.; Guyon, I.; Kasaei, S.; and Escalera, S. 2017. A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences. In *FG*, 476–483.

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Niebles, J. C. 2017. Sst: Single-stream temporal action proposals. In *CVPR*, 6373–6382.

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*, 1130–1139.

Dan, O.; Verbeek, J.; and Schmid, C. 2013. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In *ICCV*, 1817–1824.

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B. 2016. Daps: Deep action proposals for action understanding. In *ECCV*, 768–784.

Fabian Caba Heilbron, Victor Escorcia, B. G.; and Niebles, J. C. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 961–970.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R. 2017. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 3628–3636.

Girshick, R. 2015. Fast R-CNN. In *ICCV*, 1440–1448.

Huang, L.; Huang, Y.; Ouyang, W.; Wang, L.; et al. 2020. Relational Prototypical Network for Weakly Supervised Temporal Action Localization. In *AAAI*.

Jiang, Y.; Liu, J.; Zamir, A. R.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (visited on 05/12/2020).

Kang, S. M.; and Wildes, R. P. 2016. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906* .

Lee, P.; Uh, Y.; and Byun, H. 2020. Background Suppression Network for Weakly-Supervised Temporal Action Localization. In *AAAI*, 11320–11327.

Li, J.; Liu, X.; Zong, Z.; Zhao, W.; Zhang, M.; and Song, J. 2020. Graph Attention Based Proposal 3D ConvNets for Action Detection. In *AAAI*, 4626–4633.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 3889–3898.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *ECCV*.

Liu, D.; Jiang, T.; and Wang, Y. 2019. Completeness Modeling and Context Separation for Weakly Supervised Temporal Action Localization. In *CVPR*.

Liu, Y.; Ma, L.; Zhang, Y.; Liu, W.; and Chang, S.-F. 2019a. Multi-granularity generator for temporal action proposal. In *CVPR*, 3604–3613.

Liu, Z.; Wang, L.; Zhang, Q.; Gao, Z.; Niu, Z.; Zheng, N.; and Hua, G. 2019b. Weakly Supervised Temporal Action Localization through Contrast based Evaluation Networks. In *ICCV*.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 6752–6761.

Nguyen, P. X.; Ramanan, D.; and Fowlkes, C. C. 2019. Weakly-supervised action localization with background modeling. In *ICCV*, 5502–5511.

Paul, S.; Roy, S.; and Roy-Chowdhury, A. K. 2018. W-TALC: Weakly-supervised Temporal Activity Localization and Classification. In *ECCV*, 588–607.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6): 1137–1149.

Shou, Z.; Gao, H.; Zhang, L.; Miyazawa, K.; and Chang, S.-F. 2018. AutoLoc: Weakly-supervised Temporal Action Localization in Untrimmed Videos. In *ECCV*, 154–171.

Shou, Z.; Wang, D.; and Chang, S. 2016. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *CVPR*, 1049–1058.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 4325–4334.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: region convolutional 3d network for temporal activity detection. In *ICCV*, 5794–5803.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020a. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *CVPR*, 10156–10165.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020b. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *CVPR*, 10156–10165.

Yu, T.; Ren, Z.; Li, Y.; Yan, E.; Xu, N.; and Yuan, J. 2019. Temporal structure mining for weakly supervised action detection. In *ICCV*, 5522–5531.

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

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph convolutional networks for temporal action localization. In *ICCV*, 7094–7103.

Zhang, C.; Xu, Y.; Cheng, Z.; Niu, Y.; Pu, S.; Wu, F.; and Zou, F. 2019. Adversarial Seeded Sequence Growing for Weakly-Supervised Temporal Action Localization. In *ACM MM*, 738–746.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal Action Detection with Structured Segment Networks. In *ICCV*, 2933–2942.