METAL: Minimum Effort Temporal Activity Localization in Untrimmed Videos

Da Zhang[†], Xiyang Dai[‡], and Yuan-Fang Wang[†]

[†]University of California, Santa Barbara; [‡]Microsoft
{dazhang, yfwang}@cs.ucsb.edu, xiyang.dai@microsoft.com

Abstract

Existing Temporal Activity Localization (TAL) methods largely adopt strong supervision for model training which requires (1) vast amounts of untrimmed videos per each activity category and (2) accurate segment-level boundary annotations (start time and end time) for every instance. This poses a critical restriction to the current methods in practical scenarios where not only segment-level annotations are expensive to obtain but many activity categories are also rare and unobserved during training. Therefore, ***Can we learn a TAL model under weak supervision that can localize unseen activity classes?*** *To address this scenario, we define a novel example-based TAL problem called Minimum Effort Temporal Activity Localization (METAL): Given only a few examples, the goal is to find the occurrences of semantically-related segments in an untrimmed video sequence while model training is only supervised by the video-level annotation. Towards this objective, we propose a novel Similarity Pyramid Network (SPN) that adopts the few-shot learning technique of Relation Network and directly encodes hierarchical multi-scale correlations, which we learn by optimizing two complimentary loss functions in an end-to-end manner. We evaluate the SPN on the THUMOS'14 and ActivityNet datasets, of which we rearrange the videos to fit the METAL setup. Results show that our SPN achieves performance superior or competitive to state-of-the-art approaches with stronger supervision.*

1. Introduction

TAL is a fundamental problem in computer vision and has drawn increasing interests over the past few years due to its vast potential applications in security surveillance, robotics, etc. While impressive progress has been made [34, 12, 42, 6, 43, 28, 47, 7, 3, 20, 8, 31, 55, 53, 33, 48, 52, 54] to recognize and localize temporal segments in videos, success of these deep learning models heavily relies on the availability of a huge amount of labeled training data, meaning that model training requires the full annotation of the ground truth segment-level boundary for each activity instance among all possible classes. This severely limits their (1) scalability in practical scenarios as annotating temporal boundaries for long untrimmed videos is very expensive and time-consuming [46] and (2) applicability to newly emerging or rare events which are not observed in the original training dataset.

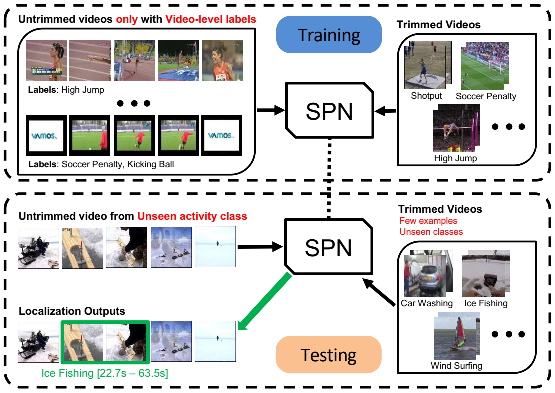

Figure 1: **Minimum Effort Temporal Activity Localization (METAL)**: during training, we simply have untrimmed videos with only video-level labels and trimmed videos of the same label; during testing, the learned model is applied to TAL in untrimmed videos given only a few trimmed examples from unseen classes.

By contrast, human beings are capable of recognizing and localizing new activity classes in untrimmed videos by observing a few examples from each class. This motivates us to develop TAL methods that require significantly fewer annotations for training and generalize well to rare and novel activity categories. In this paper, we introduce a new challenging example-based TAL problem called **Minimum Effort Temporal Activity Localization (METAL)**. As illustrated in Figure 1, we focus on the following scenario: during training, we have (1) untrimmed videos with only video-level labels (e.g. video tags) and (2) trimmed examples of the same labels, which are much easier to collect compared to segment-level boundary annotations. During testing, given only a few trimmed examples from unseen activity classes, we aim to localize all occurrences of semantically-related segments in the untrimmed testing videos. We refer this scenario as the METAL setup that this paper works on.

The METAL setup would greatly reduce the human efforts in developing efficient and scalable TAL methods and better simulate real-world scenario. To tackle this problem, we adopt the few-shot learning technique of Relation Network [40] and propose a novel meta-learning based framework, called **Similarity Pyramid Network (SPN)**. The main idea of SPN is a *hierarchical multi-scale feature representation (similarity pyramid)* that directly measures partial similarities between an untrimmed video and trimmed examples at different temporal resolutions. To train the SPN with only video-level labels, we devise two complimentary loss functions: (1) Pair-wise Content Similarity Loss (PCSL)[1] for *classification* where we compute a video-level distance metric for each pair and enforce higher similarities for positive pairs; and (2) Co-pair Structure Similarity Loss (CSSL) for *localization*, which is based on the intuition that two positive pairs should have similar distribution of similarity scores, namely higher correlation between two similarity pyramids. Thereafter, we jointly minimize the two loss functions to train the network in an end-to-end manner. The learned model is directly applied to testing videos, where the similarity pyramids are fused to yield the localization results.

Our contributions are summarized as follows:

- We introduce the METAL problem that addresses the novel task of localizing unseen activity instances in untrimmed videos given a few trimmed examples while training is only supervised by video-level labels.

- We propose a meta-learning based approach named SPN to tackle the METAL problem, which is able to measure hierarchical multi-scale similarity metrics between video pairs and simultaneously enforce classification and localization information.

- We conduct extensive experiments on two challenging benchmarks: THUMOS'14 and ActivityNet of which we rearrange the videos to fit under the METAL setup. Experimental results show that our SPN achieves performance superior or competitive to state-of-the-art approaches with stronger supervision.

2. Related Work

Temporal Activity Localization. TAL is the task to predict the temporal boundary and the label of activity instances in untrimmed videos. Earlier works on activity localization mainly used temporal sliding windows as candidates and trained activity classifiers on hand-crafted features [25, 26, 14, 16, 23, 41]. With the recent advances of deep learning methods, Conv3D network [42], two-stream convolutional networks [34, 12], and other deep neural networks [6, 43, 28, 47] have been widely applied for temporal motion analysis and significantly improved recognition performance. To localize temporal boundaries, a large body of work incorporated deep networks into the localization framework and obtained improved performance [7, 20, 8, 31, 55, 33, 48, 52, 4, 15, 11, 21, 50, 51]: Some of them focused on designing better temporal proposal schemes [4, 15, 11, 21], while others worked on improving temporal search [50, 51] or proposing better classifiers [31]. Among these works, R-C3D [48] proposed an end-to-end trainable activity detector based on Faster-RCNN [29], while S^{3}D [52] performed single-shot activity localization to get rid of temporal proposals. However, all these methods were proposed for the fully supervised setting where the segment-level boundary annotations are required during training.

Weakly Supervised TAL. Weakly supervised learning has been extensively studied for object detection [2, 10, 37]. As for activity localization, video-level label is one kind of weak supervision and has been studied in recent years. Sun *et al*. [39] was the first to consider this problem and leveraged additional supervision from web images. Hide-and-Seek [35] addressed the challenge that weakly supervised detection models usually neglect some relevant parts of the target instance. UntrimmedNet [45] proposed a framework consisting of a classification module to perform action classification and a selection module to detect important temporal segments. Most recently, AutoLoc [32] and W-TALC [27] introduced novel loss functions to further improve the performance. Although these works are trained with weak supervision, the learned models can only localize activity categories observed in the training dataset.

Few-shot Learning. Few-shot learning refers to learning from just a few training examples per class. An increasingly popular solution for few-shot learning is meta-learning where transferable knowledge can be learned from auxiliary tasks to help with the target few-shot problem. The successful MAML approach [13] aimed to meta-learn an initial condition that is good for fine-tuning on few-shot problems. To avoid fine-tuning, some works leverage the neural networks with memories [24, 30]. Another category of approach is metric-learning which aims to learn a set of projection functions such that when represented in this embedding, inputs are easy to recognize through similarity

[1] In this paper, a positive pair is defined as an untrimmed video and a trimmed video sharing the same label, while a negative pair is defined to have different labels.

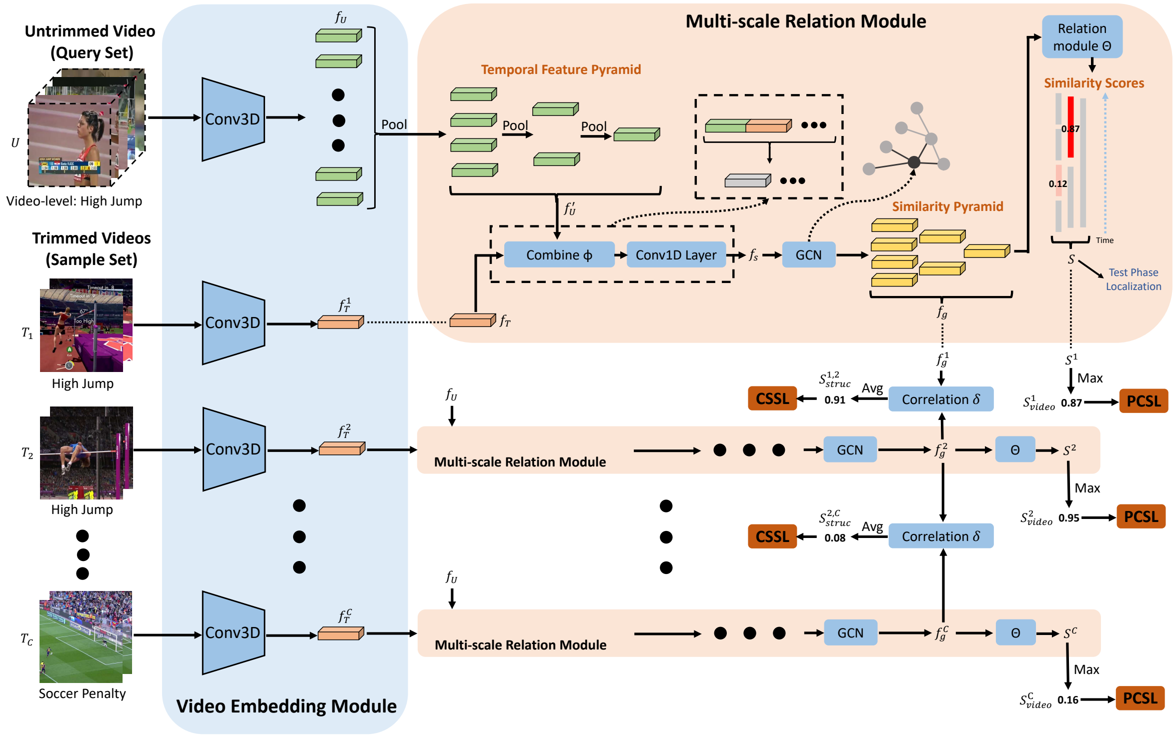

Figure 2: Similarity Pyramid Network (SPN) architecture for METAL under one-shot setting (best viewed in color). Both untrimmed and trimmed videos are fed into a shared Conv3D network for feature extraction, and a temporal feature pyramid is applied to summarize the untrimmed video. The features are then passed through the multi-scale relation module to obtain the similarity pyramids and similarity scores. Using these outputs, we compute two loss functions namely CSSL and PCSL, which are optimized jointly to train the network.

matching [19, 36, 40, 44]. While [19, 36] applies a fixed nearest-neighbor or linear classifier, [40] proposes to use a learnable non-linear function and demonstrates improved accuracy. Yang *et al*. [49] is the first work proposing the few-shot TAL task. It applied a sliding window approach with matching network to retrieve activity instances at each location. However, they still need the expensive boundary annotations to supervise the model training.

Our work is the fisrt to study the METAL problem which can also be framed as a joint problem of weakly supervised TAL and few-shot TAL, while previous works only consider one aspect at a time thus cannot be applied or easily extended to tackle the more challenging METAL setting.

3. Approach

We consider the METAL problem: Given only a few examples from unseen activity classes, the goal is to find the occurrences of semantically-related segments in an untrimmed video sequence while model training is only supervised by the video-level annotation. The setting is worth exploring as it aligns well with the practical situation: one may expect to train a localization model on dataset of easily collecting video-level labels and deploy the model to localize new activities with a few trimmed examples.

Following the few-shot learning terminologies [40, 49], we formally define the problem setup. We have three datasets: a training set, a support set and a testing set where the training set contains both untrimmed and trimmed videos with video-level labels, the support set contains labelled trimmed videos and the testing set contains untrimmed videos. The support set and testing set share the same label space, but the training set has its own label space that is disjoint with the support and testing sets. If the support set contains K trimmed examples for each of C unique classes, the target problem is called C-way K-shot.

We follow the meta-learning framework to use the *training set* during training phase and the *support set* and *testing set* during testing phase. More specifically, we follow [44, 40] to exploit the training set to mimic the few-shot learning setting via episode based training. In each training iteration, an episode is formed by randomly selecting C classes from the training trimmed videos with K samples from each of the C classes to act as the *sample set*, as well as one training untrimmed video to serve as the *query* *set*. This sample/query set split is designed to simulate the support/test set that will be encountered at test time. In our experiments (Section 4), we consider five-way one-shot ($C = 5$, $K = 1$) and five-way five-shot ($C = 5$, $K = 5$) settings.

3.1. Model Overview

In this section, we present our Similarity Pyramid Network (SPN) for METAL. An overview of our proposed SPN is illustrated in Figure 2. First, we present the video embedding module (Section 3.2) that uses a shared Conv3D network to encode both untrimmed and trimmed videos, followed by a temporal feature pyramid (Section 3.3) to naturally summarize an untrimmed video at different temporal locations and scales. We then present the multi-scale relation module (Section 3.4) that directly measures the segment-level similarities between an untrimmed video and trimmed examples. Thereafter, we introduce two loss functions PCSL and CSSL (Section 3.5), which we jointly optimize to learn the weights of the network. It may be noted that we compute both the loss functions using only the video-level labels. Finally, we show that the trained model can be directly applied for TAL given a few labelled examples in the support set (Section 3.6).

3.2. Video Embedding Module

In our problem setup, our SPN takes two types of input videos, namely, untrimmed video U and trimmed video T. We denote a video as a series of RGB frames $\{I_i\}_{i=1}^{F}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the i-th input frame and F is the total number of frames for a single video. A common practice for video processing is to use a high-quality video encoding network to extract a compact feature representation from raw frame inputs. In this work, we adopt the Res3D [43] model to obtain visual representations for both untrimmed and trimmed videos. The network weights are shared among the two different inputs.

As illustrated in Figure 2, the input RGB frame sequence can be represented as a tensor with dimension $\mathbb{R}^{F\times H\times W\times 3}$ where H and W are the height and width of each frame. For a trimmed video, we follow the traditional use of Res3D to uniformly sample L_T frames and obtain a fixed-dimensional 1D feature vector $f_T \in \mathbb{R}^{d_T}$ as the visual representation, where d_T is the number of output channels. For an untrimmed video, as the Res3D network can take arbitrary number of frames as input due to the fully convolutional nature, we also uniformly sample a much longer sequence of L_U frames and extract a feature map $f_U \in \mathbb{R}^{T_U\times d_U}$ as the visual representation where T_U is determined by the equivalent temporal stride of the original Res3D network, and d_U is the number of output channels. In the C-way one-shot setting, we feed each trimmed video to the Res3D network thus generate C features for trimmed videos. For C-way K-shot where $K > 1$, we follow [40] to element-wise sum over the Res3D outputs of all samples from each class to form this class' feature representation. Thus the number of features for the sample/support set is always C in both one-shot or few-shot setting.

After the video embedding module, we extract features for both untrimmed and trimmed videos which we denote as f_U and $\{f_T^i\}_{i=1}^{C}$ where $f_T^i \in \mathbb{R}^{d_T}$ represents each class' feature. Note that $\{f_T^i\}_{i=1}^{C}$ are from C different classes during testing but not necessarily in the sample set (during training) in order to enrich the training dynamics.

3.3. Temporal Feature Pyramid

Although f_U serves as a good feature representation for an untrimmed video, it only summaries the video at a single temporal resolution. One may think of applying the temporal sliding window approach [49], but such method is computationally intensive and cannot model complex temporal relations. Inspired by the single-shot object detector [22] and its successful applications in temporal activity localization [52, 20], we construct a multi-scale feature pyramid to directly produce temporal features at variable scales. Unlike the previous activity localization methods trained with strong supervision, the few-shot problem setup requires us to minimize the network size to prevent overfitting. Thus, we use a simple multi-scale pooling architecture instead of multiple layers of temporal convolutions.

Specifically, we stack N_U 1D max-pooling layers with a pooling stride of 2 to generate a sequence of feature maps that progressively decrease in temporal dimension which we denote as $\{f_U^i\}_{i=1}^{N_U}$, $f_U^i \in \mathbb{R}^{T_U^i\times d_U}$ where T_U^k is the temporal dimension of each layer. Thus each temporal feature is responsive to a particular temporal location and scale. For simplicity, we denote the final encoding feature for an untrimmed video as $f'_U \in \mathbb{R}^{N\times d_U}$ where $N = \sum_{i=1}^{N_U} T_U^i$ is the total number of temporal locations used for the multi-scale feature pyramid.

3.4. Multi-scale Relation Module

To learn the relations between untrimmed and trimmed videos, we follow the relation network [40] to combine the feature maps between two different inputs with operator $\Phi(f'_U, f_T)$, where f_T is a class' feature map and we omit the superscript for simplicity. Different from the relation network where only image-to-image relations are considered, we extend the formulation to video domain and deal with relations between untrimmed and trimmed videos. In this work, we assume $\Phi(\cdot,\cdot)$ to be concatenation of feature maps in depth among all temporal locations defined as:

$$f_\Phi = \Phi(f'_U, f_T) \in \mathbb{R}^{N\times d_\Phi} \tag{1}$$

where $d_\Phi = d_U + d_T$ is the number of channels after concatenation. We then generate a similarity embedding f_s using one single 1D convolutional (Conv1D) layer:

$$f_s = ReLU(Conv1D(f_\Phi)) \in \mathbb{R}^{N \times d_s} \tag{2}$$

where d_s is the number of output channels.

While f_s can be directly fed into a relation module to compute the similarity scores, it only considers the content similarity at each specific temporal location. However, temporal contextual information has been proven to be critical for TAL [7, 54, 53]. To encode such contextual relations in our network, we adopt a simple GCN on top of f_s. Different from the standard convolutions which operate on a local regular grid, the graph convolutions allow us to compute the response of a node based on its neighbors defined by the graph connections. In this work, temporal segments are represented by nodes, and their relations are defined as edges. We use f_s as the input node features and one layer of graph convolution is defined as:

$$f_g = ReLU(Gf_sW) \tag{3}$$

where $G \in \mathbb{R}^{N \times N}$ is the adjacency matrix, f_s is the input feature of all nodes, $W \in \mathbb{R}^{d_s \times d_g}$ is the learnable weight matrix and $f_g \in \mathbb{R}^{N \times d_g}$ is the output node representation. In this work, we define the adjacency matrix based on the ordering of temporal segments as originally encoded in the multi-scale feature hierarchy. After one GCN layer, each node representation in f_g is enriched by the neighborhood relations. We refer to f_g as the similarity pyramid as it naturally encodes relations in a multi-scale feature pyramid.

Finally, we apply a relation module $\Theta(f_g)$ to produce similarity scores $S \in \mathbb{R}^N$ where each number is a scalar in range of 0 to 1 representing the similarity at each temporal location. In this work we assume $\Theta(\cdot)$ be a multi-Conv1D layer although other choices are possible.

3.5. Training

In this section, we present two proposed loss functions which use only the video-level labels as direct supervision for classification and localization, respectively. To better illustrate our idea, we consider one training batch containing one untrimmed feature f_U and C trimmed features $\{f_T^i\}_{i=1}^C$.

Pair-wise Content Similarity Loss. Here, we propose a Pair-wise Content Similarity Loss (PCSL) to add classification constraints. Considering one positive pair, although we don't know which temporal segment best corresponds to the trimmed example, it is certain that there is at least one semantically-related segment resulting in a high similarity score (close to 1). Similarly, all similarity scores will be small (close to 0) considering a negative pair. Based on this motivation, we aggregate similarity scores S to form a video-level score S_{video} via a simple max-pooling. Given a pair (f_U, f_T^i), S_{video}^i will be regressed to 1 if it is positive, otherwise 0.

Given the labels of untrimmed and trimmed videos in one batch, we formally define a positive set S_p containing all positive pairs and a negative set S_n where $|S_p| + |S_n| = C$. We define the PCSL as the sum of the sigmoid cross entropy loss for each pair:

$$\mathcal{L}_{PCSL} = -\sum_{i=1}^{C} \mathcal{L}_{sigmoid}(S_{video}^i, GT_{video}^i) \tag{4}$$

where S_{video}^i is the predicted video-level score, GT_{video}^i is the ground truth score $GT_{video}^i = 1, (f_U, f_T^i) \in S_p$ and $GT_{video}^i = 0, (f_U, f_T^i) \in S_n$.

Co-pair Structure Similarity Loss. While PCSL enforces the pair-wise relations between untrimmed and trimmed videos, it is location agnostic as it only measures the video-level similarity. In order to provide constraints for learning better weights for localization, we propose another Co-pair Structure Similarity Loss (CSSL). Our intuition is that given two positive pairs, for example an untrimmed video of playing basketball and two different trimmed videos of shooting, both should be matched to the same temporal region in the untrimmed sequence although the boundary annotation is unknown. To enforce such information during training, we leverage the design of similarity pyramid f_g and enforce two pyramids to have similar structures (distribution of scores) for two positive pairs.

Formally, given two positive pairs (f_U, f_T^a) and (f_U, f_T^b), we first produce the similarity pyramid after GCN as f_g^a and f_g^b respectively. Based on the above intuition, we compute the structure similarity between two similarity pyramids. Specifically, we define the structure similarity as the average cosine similarity among all temporal locations:

$$\begin{aligned} S_{struc}^{a,b} &= \frac{1}{N}\sum_{i=1}^{N} \delta(f_g^a(i), f_g^b(i)) \\ \delta(f_g^a(i), f_g^b(i)) &= \frac{(f_g^a(i))^T f_g^b(i)}{||f_g^a(i)|| \cdot ||f_g^b(i)||} \end{aligned} \tag{5}$$

where $f_g^a(i)$ and $f_g^b(i)$ indicates the feature vector at index i and $\delta(\cdot,\cdot)$ denotes the cosine similarity between two features. Note that the embeddings f_g^a and f_g^b are multi-scale similarity embeddings among different temporal locations, thus the score S_{struc} peaks when they share the same distribution. Therefore, given one positive pair, S_{struc} will be maximized when compared with another positive pair, otherwise minimized.

Given a training batch, we define the CSSL as the sum of structure similarities for every two pairs including at least one positive pair:

$$\mathcal{L}_{CSSL} = \sum_{i=1}^{|S_p|}\sum_{j=1}^{|S_n|} S_{struc}^{i,j} - \sum_{i=1}^{|S_p|}\sum_{j=i+1}^{|S_p|} S_{struc}^{i,j} \tag{6}$$

where $S^{i,j}_{struc}$ is the predicted structure similarity, $|S_p|$ is the number of positive pairs and $|S_n|$ is the number of negative pairs.

Finally, the SPN is end-to-end trainable by jointly optimizing two loss functions. The joint training allows all network weights to be trained such that the embedding module as well as the relation module are optimized for both classification and localization. The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{PCSL} + \alpha \mathcal{L}_{CSSL} \tag{7}$$

where α is used to balance the two losses.

3.6. Prediction

TAL via SPN is straightforward with one forward pass of the network. Considering a C-way K-shot localization problem with one untrimmed testing video and K different trimmed videos in each of the C different classes from the support set. We first extract the visual features for both untrimmed videos and trimmed videos resulting in C trimmed features and 1 untrimmed feature. Then, we compute, as the outputs of multi-scale relation module, the similarity scores S for each of the C features. For each specific temporal location, the maximum similarity score among C different classes and the corresponding class label are assigned for the temporal segment. Then the segments with similarity score less than 0.5 will be filtered out and the remaining segments are refined via temporal non-maximum suppression to get the final localization results.

4. Experiments

In this section we describe the experimental results of our method. First, we introduce the evaluation settings for the METAL setup and the implementation details of our model. Then we compare our SPN with other state-of-the-art approaches. Finally, we perform ablation studies to investigate the impact of different components of our approach and provide qualitative visualizations.

4.1. Datasets and Evaluation Settings

We evaluate our SPN on two large-scale datasets, namely THUMOS'14 [17] and ActivityNet [5]. While the original datasets are collected for TAL with strong supervision, we rearrange the videos to fit under the METAL setup by (1) Removing the boundary annotations for untrimmed videos; (2) Splitting activity classes into mutually exclusive sets; (3) Pairing each untrimmed video with trimmed examples from different sources. We detail the evaluation settings below.

Evaluation settings. We follow the problem definition as described previously. In our experiments, we consider the five-way localization problem under one-shot ($K = 1$) and five-shot ($K = 5$) settings. During the training, in each iteration, we construct the *sample set* by randomly sampling five classes from the subset of the training classes, and then for each class we randomly sample K trimmed videos. For the *query set*, we randomly sample one untrimmed video. During the testing phase, the setup is identical to that of the training phase, only now we use the *support set* and *testing set*. Note that the support set should have at least one class overlap with the video-level label in the testing set.

We follow the conventions to report the mean Average Precision - mAP@a where a denotes the temporal Intersection over Union (tIoU) threshold, and the average mAP among 10 tIoU thresholds [0.5:0.05:0.95]. As can be easily seen, there are a large number of different combinations of the trimmed and untrimmed videos (random classes and random samples), and the performance is dependent on those choices. We follow the few-shot tradition [49, 40] to get the reliable test results, namely, we randomly sample 1000 testing batches and the final results are reported by averaging over all these batches.

ActivityNet v1.2 [5] ActivityNet is a recently released benchmark for temporal activity localization. The dataset is released in two versions, and to facilitate comparisons with previous works, we use the version 1.2 which contains 4819 and 2383 untrimmed videos in the original training and validation subsets respectively. There are 100 different activity classes and we randomly split it into 80 classes (ActivityNet-train-80) for training and 20 classes (ActivityNet-test-20) for testing. We use the video segments in ActivityNet as the trimmed samples and we make sure that trimmed videos do not come from the same untrimmed video when pairing them together.

THUMOS'14 [17] The THUMOS'14 dataset is another widely used benchmark for activity recognition and localization. There are 2765 trimmed videos from UCF101 dataset [38] and 413 untrimmed videos of 20 different activity categories. Although this is a smaller dataset, it has several videos where multiple activities occur, thus making it even more challenging. The 20 classes are a subset of the 101 classes in UCF101. Following [49], we split the 20 classes into 6 classes for training and 14 classes for testing. The two splits are denoted as Thumos-train-6 and Thumos-test-14. The trimmed videos come from mutual classes in UCF-101 which we denote as UCF-101-6 and UCF-101-14 for training and testing respectively.

4.2. Implementation Details

For the video embedding module, we train a Res3D model [43] on the Kinetics dataset [6]. Note that the few-shot problem setup requires that the classes for testing must not be present during training and we notice that there are mutual classes between Kinetics and ActivityNet or THUMOS'14, thus, those classes are excluded when we train the Res3D model. As described in Section 3.2, we set $L_T = 24$, $L_U = 256$ and $d_T = d_U = 2048$. On THUMOS'14, as the

Method	Supervision	Few-shot	mAP@0.5		average mAP	
			1-shot	5-shot	1-shot	5-shot
CDC [31]	Full	Yes	8.2	8.6	2.4	2.5
Yang et al. [49]	Full	Yes	22.3	23.1	9.8	10.0
SPN (ours)	**Weak**	**Yes**	**41.9**	**45.0**	**26.5**	**28.8**
AutoLoc [32]	Weak	No	45.2		30.8	

Table 1: TAL results on ActivityNet v1.2 (in percentage). mAP at tIoU threshold 0.5 and average mAP are reported. Methods are categorized into three groups: Weak supervision provides video-level labels during training; Full supervision provides temporal boundary annotations during training; Few-shot refers to only a few labeled examples are available.

Method	mAP@0.5	
	1-shot	5-shot
CDC [31]	6.4	6.5
Yang et al. [49]	13.6	14.0
SPN (ours)	**14.3**	**16.2**
AutoLoc [32]	24.5	

Table 2: TAL results on THUMOS'14 (in percentage). mAP at tIoU threshold 0.5 is reported. The methods are categorized into the same groups as used in Table 1.

length of untrimmed videos is much longer, we follow common practice [48] to cut it into non-overlapping 32-second segments and use the segmented inputs. Regarding the temporal feature pyramid, we use $N_U = 5$ for ActivityNet to generate a sequence of feature maps with temporal dimension $\{16, 8, 4, 2, 1\}$ and $N_U = 3$ for THUMOS'14 to produce the features maps with temporal dimension $\{16, 8, 4\}$. We set $d_s = d_g = 512$ for the multi-scale relation module, and the relation module $\Theta(\cdot)$ is two layers of Conv1D to map feature input to similarity scores with sigmoid activation. The whole SPN network is optimized with the end-to-end loss function defined in Equation 7. As a speed accuracy trade-off, only the last layer of the Res3D model is jointly optimized after pre-training. We implement our SPN on TensorFlow [1]. The whole network is trained by Adam [18] optimizer with learning rate 10^{-5}.

4.3. Comparison with State-of-the-art

As there are no existing methods for TAL under the METAL setup. we make comparisons with state-of-the-art localization models trained with *stronger* supervision. Specifically, we compare with the methods which are trained with video-level labels but not under few-shot settings [32][2], and the methods proposed for few-shot activity localization but trained with temporal boundary annotations [31, 49][3]. It should be emphasized again that results of our methods are reported under the METAL setting which is most challenging of all.

Method	mAP@0.5	average mAP
Yang et al. [49]	22.3	9.8
SPN-ImageNet	35.2	20.6
SPN-Kinetics	**41.9**	**26.5**
Base	13.2	7.2
+Feature Pyramid	30.3	18.2
+GCN	34.7	22.7
+CSSL	**41.9**	**26.5**

Table 3: Ablation study for different SPN components on ActivityNet. Top: Weight initialization for the embedding module. Bottom: Effectiveness of temporal feature pyramid, GCN and CSSL. Results are reported under five-way one-shot localization.

ActivityNet v1.2 Table 1 shows the localization results on the ActivityNet v1.2 dataset. All the methods are categorized into three different groups based on the level of supervision. Our SPN under the one-shot setting, significantly outperforms previous fully supervised methods among all evaluation metrics, demonstrating the superior ability of our model to effectively learn good similarity metrics between different video pairs even without having access to boundary annotations. Compared to the weakly supervised method trained with more data, although our method lacks in performance for one-shot localization, we achieves competitive accuracy when more labelled data are available (*i.e.* five-shot localization). It should be noted that we still use fewer annotations compared to that of those used in [32].

THUMOS'14 We also compare our method with the state-of-the-art approaches on THUMOS'14 dataset. The results are shown in Table 2 where the methods are also categorized into the same groups as used in Table 1. Our SPN consistently achieves superior or competitive performance compared with previous methods trained with stronger supervision. Note that THUMOS'14 is a more challenging dataset than ActivityNet for the METAL problem, as the former has much longer untrimmed videos and has more activity instances per video, making it harder to efficiently model similarities under weak supervision: on average, the THUMOS'14 training set has 15 instances per video, while the ActivityNet training set has only 1.5 instances per video.

[2] Results are reported using the few-shot evaluation settings.

[3] For CDC, we use the values reported in [49]

Hence, strong adaptivity is required to perform consistently well on both datasets.

4.4. Ablation Studies

Weight Initialization. We conduct experiments to study the effect of different weight initialization for the embedding module. We consider two different initialization: (1) Res3D initialized from ImageNet [9] weights (simply duplicate 2D kernels to 3D) without pre-training on any video datasets, we denote as SPN-ImageNet. (2) Res3D pretrained from Kinetics (Section 4.2), we denote as SPN-Kinetics. The results are summarized in the top half of Table 3. It may be noted that our SPN-ImageNet already significantly outperforms the state-of-the-art method, highlighting SPN's strong ability to learn the temporal relations.

Network Components. On ActivityNet v1.2 dataset, we perform ablation studies to investigate the effect of each network component we proposed in this paper: temporal feature pyramid and GCN. All the experiments are conducted for five-way one-shot localization.

First, we implement a baseline model: we use the same Res3D network to extract features for both the untrimmed video and trimmed videos, instead of using a multi-scale architecture to encode the untrimmed video, we directly apply a relation module to compute 32 relation scores which is then max-pooled and trained with video-level labels (PCSL only). As each score only represents a small duration of the entire video, we apply multi-scale sliding windows and use the maximum score for each windowed segment. The result is reported in the first row in bottom half of Table 3.

On top of this base model, we first add the temporal feature pyramid and leave other parts unchanged to study the effect of this component alone. The result is shown in the second row in bottom half of Table 3. We observed a significant performance jump improving mAP@0.5 from 13.2 to 30.3, this clearly demonstrates the advantage of using a multi-scale feature pyramid to directly summarize video content at different temporal locations and scales.

We further validate our design to use a GCN for modeling contextual relations in the multi-scale relation module. Specifically, based on the previous model, we add a GCN on top. As reported in the third row in bottom half of Table 3, we achieve higher mAP indicating the importance to enrich similarity by contextual relations.

CSSL. One major contribution of SPN is to add a CSSL during training to enforce localization supervision even without boundary annotations. As shown in the Table 3, adding the CSSL improves the mAP@0.5 from 34.7 to 41.9 and average mAP from 22.7 to 26.5. This significant improvement indicates the importance of training SPN with CSSL and supports our motivation to enforce structure similarity between two video pairs.

Qualitative Visualization. As shown in Figure 3, We provide further qualitative visualization of the similarity scores. Although the untrimmed video and trimmed examples differ a lot in terms of motion and appearance, our SPN can output higher scores for more related segments and only keep the best matched ones through temporal NMS, demonstrating the effectiveness and robustness of the proposed framework.

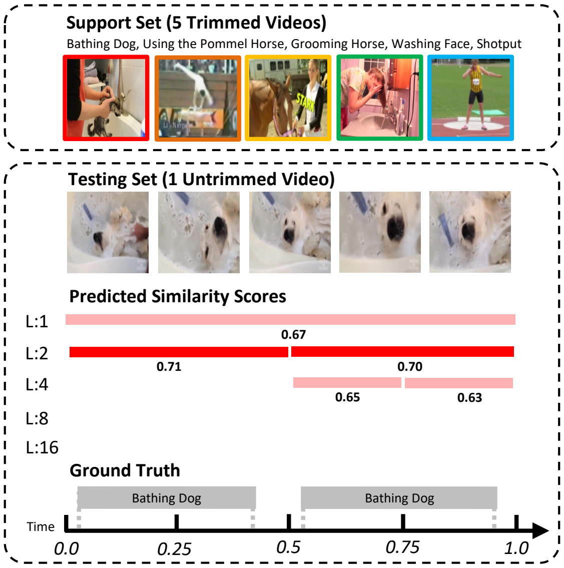
Support Set (5 Trimmed Videos)
Bathing Dog, Using the Pommel Horse, Grooming Horse, Washing Face, Shotput
Testing Set (1 Untrimmed Video)
Predicted Similarity Scores
L:1
0.67
L:2
0.71
0.70
L:4
0.65
0.63
L:8
L:16
Ground Truth
Bathing Dog
Bathing Dog
Time
0.0
0.25
0.5
0.75
1.0

Figure 3: Qualitative Visualization of the similarity scores in ActivityNet v1.2 dataset (best viewed in color). The segments with top 5 scores are visualized with each class in the support set shown in different colors. The predicted segments are organized with different temporal resolutions, and the similarity score is shown below each segment. Light color indicates that the corresponding segment is suppressed by temporal NMS. For better visualization, the temporal length of the video is normalized to 1.0.

5. Conclusion

In this paper, we introduce a new challenging setting for TAL in untrimmed videos called Minimum Effort Temporal Activity Localization (METAL) which can also be framed as a joint problem of weakly supervised and few-shot TAL. We have presented SPN, a Similarity Pyramid Network that adapts a meta-learning framework to address the challenges in a single shot end-to-end architecture. Given only video-level labels, our SPN is end-to-end trainable by optimizing two complimentary loss functions and generalizes well to localize unseen activity classes. With this framework, although trained under the METAL setup on the challenging THUMOS'14 and ActivityNet benchmarks, our SPN achieves performance superior or competitive to that of those state-of-the-art approaches with stronger supervision.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 7

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 2

[3] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. 1

[4] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016. 2

[5] 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*, pages 961–970, 2015. 6

[6] 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*, pages 6299–6308, 2017. 1, 2, 6

[7] 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

[8] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5727–5736. IEEE, 2017. 1, 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[10] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 914–922, 2017. 2

[11] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016. 2

[12] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 1, 2

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 2

[14] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013. 2

[15] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3628–3636, 2017. 2

[16] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 740–747, 2014. 2

[17] YG Jiang, J Liu, A Roshan Zamir, G Toderici, I Laptev, M Shah, and R Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 6

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015. 3

[20] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 988–996. ACM, 2017. 1, 2, 4

[21] 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. 2

[22] 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*, pages 21–37. Springer, 2016. 4

[23] Pascal Mettes, Jan C van Gemert, Spencer Cappallo, Thomas Mensink, and Cees GM Snoek. Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 427–434. ACM, 2015. 2

[24] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org, 2017. 2

[25] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*, pages 1817–1824, 2013. 2

[26] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Efficient action localization with approximately normalized

fisher vectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2545–2552, 2014. 2

[27] 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

[28] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017. 1, 2

[29] 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*, pages 91–99, 2015. 2

[30] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016. 2

[31] 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 *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1417–1426. IEEE, 2017. 1, 2, 7

[32] 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. 2, 7

[33] 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*, pages 1049–1058, 2016. 1, 2

[34] 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. 1, 2

[35] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017. 2

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 3

[37] Hyun Oh Song, Yong Jae Lee, Stefanie Jegelka, and Trevor Darrell. Weakly-supervised discovery of visual pattern configurations. In *Advances in Neural Information Processing Systems*, pages 1637–1645, 2014. 2

[38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[39] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. ACM, 2015. 2

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2, 3, 4, 6

[41] Kevin Tang, Bangpeng Yao, Li Fei-Fei, and Daphne Koller. Combining the right features for complex event recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2696–2703, 2013. 2

[42] 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*, pages 4489–4497, 2015. 1, 2

[43] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1, 2, 4, 6

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 3

[45] 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*, pages 4325–4334, 2017. 2

[46] 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. 1

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *CVPR*, 2018. 1, 2

[48] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 6, page 8, 2017. 1, 2, 7

[49] Hongtao Yang, Xuming He, and Fatih Porikli. One-shot action localization by learning sequence matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1450–1459, 2018. 3, 4, 6, 7

[50] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 2

[51] Gang Yu and Junsong Yuan. Fast action proposals for human action detection and search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1302–1311, 2015. 2

[52] Da Zhang, Xiyang Dai, Xin Wang, and Yuan-Fang Wang. S3d: Single shot multi-span detector via fully 3d convolutional network. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 1, 2, 4

[53] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1247–1257, 2019. 1, 5

[54] Da Zhang, Xiyang Dai, and Yuan-Fang Wang. Dynamic temporal pyramid network: A closer look at multi-scale modeling for activity detection. In *The Asian Conference on Computer Vision(ACCV)*, 2018. 1, 5

[55] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2933–2942. IEEE, 2017. 1, 2