Selective Feature Aggregation for Single Frame Supervised Temporal Action Localization

Volume I

Danqi Liao

A Master's Thesis
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Master of Science in Engineering

Recommended for Acceptance by
the Department of
Computer Science

Adviser: Olga Russakovsky

May 2022

Abstract

Action detection in untrimmed video has been a long standing goal in computer vision. Recently, single-frame annotation has emerged as a promising direction that bridges the gap between the video-level weak-supervision and costly full supervision. We tackle the problem of single-frame supervised temporal action localization, where only one frame is annotated for each action instance in the video. Contextual information is crucial for recognizing and localizing action instances. However, existing methods for single-frame action detection still rely on limited isolated features.

In this thesis, we propose the Selective Feature Aggregation module, which (1) dynamically aggregates the contextual information to strengthen the expressive power of the perframe features, and (2) utilizes a set of selective functions, which encode a general prior for selecting neighbors, to guide the feature aggregation. We find that this module reduces the context confusion and attention collapse when training a feature aggregator with a very sparse set of labels.

We demonstrate that our proposed module can effectively improve the performance over previous methods on three benchmarks: THUMOS'14, GTEA and BEOID. Concretely, we improve 3.1%, 7.9%, and 2.8% respectively in IoU-averaged mAP over the baseline SFNet. The benefits are particularly striking on the challenging setting with an IoU of 0.7, where we improve 10.8% over competitive methods on BEOID.

Acknowledgements

In the extraordinary time of a global pandemic, this thesis would not have been possible without the support of many people.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Olga Russakovsky, for her encouragement, guidance, and continuous support for my research. I could not have asked for a better advisor for my Master's studies.

I am also grateful to my collaborators, Zhiwei Deng and Felix Yu, for their motivation, help, and insightful discussion of this work. I have learned so much about research and research in computer vision from working with them.

I thank my Princeton Visual AI Lab fellows Sunnie Kim, Jihoon Chung, Yu Wu, Dora Zhao, Vikram Ramaswamy, and Angelina Wang for their support, stimulating discussion during lab meetings, and helpful feedback for this work.

Last but not the least, I would like to thank my parents, Bilian Zheng and Wentong Liao, for believing in me, supporting my education, and always being on my side. I would not have made it this far without them and their unconditional love. To them, I dedicate this work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Action detection in untrimmed videos has been a long-standing goal in computer vision. Remarkable progress has been made in fully-supervised settings [16, 10, 42, 35, 49, 4, 11, 39, 41]. However, obtaining precise annotations for temporal action localization is often prohibitively time-consuming. Researchers have resorted to weakly-supervised learning paradigms to reduce the cost [38, 40, 28, 29, 22].

Single-frame annotation emerges as a promising direction which bridges the gap between the video-level weak supervision where no temporal information about the location of action instances is provided, and full supervision where costly per-frame dense annotations are used. In the single-frame setting, one frame is annotated for each action instance in a video, providing the information about the action classes appearing in the video and their approximate location. This annotation strategy achieves a balance between the lack of enough information and the large annotation costs for complete supervisions.

With only sparsely pinpointed location information, a more powerful learning framework is required to infer and complete the missing annotations. One widely-used framework is to iteratively obtain pseudo labels [20] from a weakly-trained model, which are used as a complementary signal for further model training. However, stan-

dard pseudo label methods are designed for i.i.d. data. Directly applying the pseudo label methods on video frames loses the temporal structure information among the elements.

Exploiting the *temporal contextual information* is the key to single-frame supervised action detection. A video frame's context defines a more precise meaning of the frame, resolves the ambiguity, and contains the consistency, temporal order, and structure of an action. Existing methods work on the pseudo label space and propose various post-processing techniques to incorporate temporal contextual priors, such as consistency, into the pseudo label generation process. For example, anchor-based expansions and background mining are proposed to use the neighboring frames as the expansion target to incorporate the temporal smoothness prior [29]. To further impose a stronger contextual prior on the temporal structures of the generated pseudo labels [18], either a second-phase proposal-based refinement module or an optimal sequence search with non-maximum suppression [22] is needed. The "pseudo label with contextual post-processing" paradigm has greatly boosted the performance of single-frame temporal action localization, but mainly works with the label space and often needs to heavily rely on heuristic rules and hand-designed systems.

Another paradigm for incorporating the contextual information is to work directly with the feature space, which contains much richer information. In fact, feature learning and aggregation methods have been a key driving force in image and video understanding [48, 8, 1]. Specifically, for temporal action localization, graph neural networks [50] and transformer-based [33] architectures have been proposed to encode the long-range dependencies among proposals and frames. However, under the single-frame weakly supervised learning setting, directly applying the strong feature aggregators can potentially lead to various issues. Due to the scarcity of supervisions on the temporal locations, the attention scores learned along the temporal axis tend to suffer from the collapse phenomenon, where the attention focuses on just a small

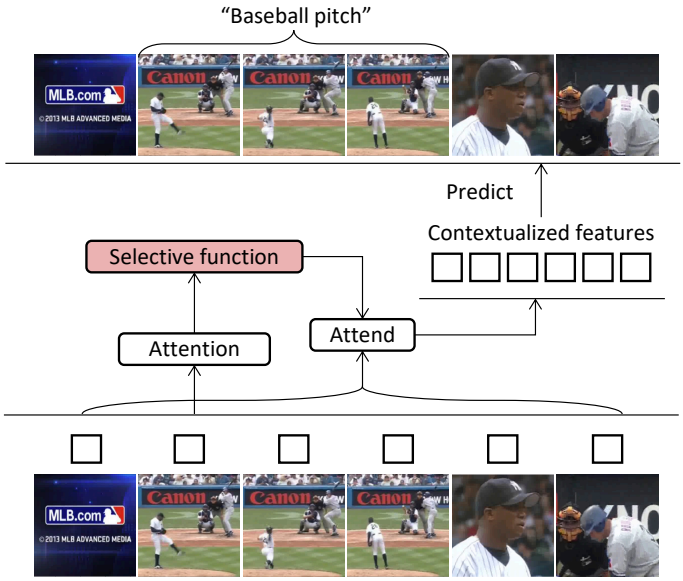

Figure 1.1: We propose the *Selective Feature Aggregation* module that selectively aggregates frame features. The module successfully utilizes temporal contextual information, leading to performance boosts in single frame-supervised temporal action localization without complex heuristic-based system designs.

handful of frames. Similar findings are reported in prior work [7, 54]. Simultaneously, building a global feature aggregation module can also lead to action-context confusion [38] in a wider range, where, due to lack of labels, contextual frames and clips are misclassified as other classes depending on how the context is collected in aggregation.

In this thesis, we propose the *Selective Feature Aggregation* module, shown in Figure 1.1, that utilizes the contextual information in a constrained way along the temporal axis. The module takes in the isolated frame-level features, dynamically attends to the relevant elements under the guidance of a selective function, and performs feature aggregation for further classification. We show that this simple module can work well with a basic pseudo label mining strategy [29] and achieve competitive performance compared to existing methods.

Concretely, our contributions are:

- We introduce a new *Selective Feature Aggregation* module, which aggregates

the information along the temporal axis in a dynamic way under the guidance of a selective function, providing a base module for the single-frame supervised temporal action localization.

- We perform a thorough analysis of the proposed module and discuss the insights on the design and behavior of the feature aggregator under the scarcity of labels. We find that through adopting a general selective prior, the learned attention can be regularized to effectively utilize context for action localization.

- We show that our method achieves competitive performance in single frame-supervised temporal action localization on three benchmark datasets (THUMOS'14 [17], GTEA [23] and BEOID [6]) without having to resort to heuristic-based designs of the label-space post-processing system. Concretely, we achieve consistent improvements over the baseline framework SF-Net [29]; we perform especially strongly on the challenging localization setting of IoU 0.7, where we improve 10.8% over the competitive method by Ju et al. [18] on BEOID.

Chapter 2

Related Work

2.1 Fully-supervised temporal action localization

In order to perform temporal action localization in videos, a standard fully-supervised setting utilizes precise annotations on both action boundaries and classes for model training. Early methods in this setting often adopt a sliding-window based proposal method [10, 28, 35, 49], where a model is trained to classify proposals at all temporal scales. To reduce the search space, several paradigms are proposed: (1) bottom-up based merging methods, where video segments are classified to obtain the action boundaries [39, 31]; (2) two-stage framework where a proposal generator is learned for further action classification and boundary refinement [41, 46, 53, 2, 13, 9, 12]; and (3) end-to-end architectures [4, 5, 11, 46, 47, 33]. Recently, there has been a trend to adopt strong feature aggregation models to perform action localization, such as graph convolutional networks [50] or transformers [33]. These methods are designed to heavily leverage the full supervision, and lack the ability to handle the scarcity of the labels in weakly supervised settings.

2.2 Weakly-supervised temporal action localization

Early weakly-supervised settings focus on utilizing video level labels for action detection. STPN [34] uses sparsity constraints to learn the key subset of segments for action localization. Untrimmednets [44] proposes to learn a selection module for ranking clips that contribute to the video classification. To model the class-agnostic information within the videos, DGAM [38] proposes to train the attention model with both generative and discriminative modules. The classical Expectation Maximization with Multi-instance Learning framework is also proposed [28] to perform action localization with video labels. Representation learning methods are also proposed for this setting [14, 36, 32]. Action Graphs [37] learns the similarity and dissimilarity between segments through graphs for segment selection. AutoLoc [40] uses an Outer-Inner-Contrastive loss for a boundary predictor.

Recently, point-level supervision gains attention to serve as a transition step between full supervision and video-level weak supervision [30, 29], where frame mining and expansion around the point-level supervision are performed to obtain more information. Ju et al. [18] proposes to combine the frame-level paradigm and proposal-level paradim for localization. Lee et al. [22] shows that learning the completeness of actions can improve the localization performance. We follow the same setting of point-level supervision and show that building a strong feature aggregation module can remedy the lack of contextual information without resorting to complex heuristic-based refinement systems.

2.3 Contextual information for recognition

Utilizing the correct contextual information is crucial for vision tasks. In semantic segmentation and object detection, feature aggregations are used to obtain both global and fine-grained details [52, 25]. Space-time feature representation learning are

adopted for video understanding [1, 45, 15]. Recently, transformers are transitioning to a dominant model for feature aggregations [8, 26, 27]. Instead of relying on the known structure among elements, full supervision or large amounts of data, we focus on the feature aggregation method when only a small portion of data is labeled, and the model needs to handle the unknown temporal structures and the potential context confusion for feature aggregation.

Chapter 3

Model

In this section, we first define the problem formulation(Section 3.1), then present the learning framework for our model (Section 3.2), and finally the proposed feature aggregation module for temporal action localization (Section 3.3).

3.1 Problem formulation

Denote each input video as $V = \{I_t\}_{t=1}^{T}$, where I_t corresponds to the frame at timestep t and T is the video length.[1] There are C action classes to be detected. For every training video, the labels consist of C sets of $\mathcal{Y}_c \subseteq \{1, \ldots, T\}$ where $t \in \mathcal{Y}_c$ indicates that frame t in the video is known to contain the action c. Thus, in our single-frame label case, $|\mathcal{Y}_c|$ will correspond to the number of instances of action c in this video. The ultimate goal is to output on a test video a set of predictions $\{(c_i, t_i^{start}, t_i^{end}, s_i)\}_i$ of class labels c_i, time intervals, and confidence scores s_i, which will then be compared against full test annotations.

[1] We use "frame" for simplicity in the text when referring to timestep t; in practice we instead use 16-frame clips as in [29].

3.2 Learning and inference framework

We begin by training a model to accurately classify each video frame using the sparse labels. Following [29], we adopt two output modules in the model: (1) the frame classification module, and (2) the actionness module. Then, we apply their post-processing approach to convert these frame-level scores into temporal interval predictions.

Feature extraction

Each video V contains a sequence of frames $(I_1, ..., I_T)$. All the frames are passed through the I3D feature extraction backbone pre-trained on Kinetics [3] to obtain the sequence of compact feature representation $(\mathbf{x}_1, ..., \mathbf{x}_T)$, $\mathbf{x}_t \in \mathbb{R}^D$, where T is the total number of video frames and D is the dimension for the feature space.

Module 1: frame classification

The frame classification module builds an action classifier for each frame in the video. Concretely, it contains a simple three-layer perceptron F that takes in the frame features $\mathbf{x}_t$ and returns class probabilities $\mathbf{p}_t \in [0, 1]^{C+1}$:

$$\mathbf{p}_t = \text{SOFTMAX}(F(\mathbf{x}_t)) \tag{3.1}$$

This vector includes a *background* probability $p_{t,0}$, since not every frame will correspond to one of C action classes.

Since there are no negative labels initially, *background mining* is performed using multiple-instance learning. Let $\mathcal{Y}^{pos} = \cup_{c=1}^{C} \mathcal{Y}_c$ be all the positive labeled frames, for any class. At every iteration of training, the K frames not in $\mathcal{Y}^{pos}$ with the highest background probability $\mathbf{p}_{:,0}$ become the background label set $\mathcal{Y}_0$. K is set following [29] to be η times the number of positive frames $|\mathcal{Y}^{pos}|$, with $\eta = 7$.

Module 2: actionnness

The second module captures the likelihood of *any* action appearing in the frame. It performs a binary classification and outputs a scalar probability score. We follow the design in [29] and use a function mapping G with two temporal convolution layer and one linear layer with ReLU as the activation function in between:

$$\mathbf{a} = \text{SIGMOID}(G([\mathbf{x}_1; ...; \mathbf{x}_T])) \tag{3.2}$$

where $G(\cdot)$ takes in the temporally concatenated features and produce the actionness scores $\mathbf{a} \in [0, 1]^T$ simultaneously for every frame in the video.

Video-level score

During training we also compute (and supervise) a video-level classification score. Concretely, we compute a probability vector $\tilde{\mathbf{p}} \in [0, 1]^{C+1}$, where for each class c, we follow common practice in e.g., [29]: identify the M highest-probability frames ($M = L/8$ for video length L) and set $\tilde{p_c}$ to be the average p_{tc} over those frames. Then we run a softmax on $\tilde{\mathbf{p}}$.

Overall training objective

We define a overall loss function, which aggregates (via a weighted sum) the losses from the different components:

$$\mathcal{L}_{frame} = -\sum_{c=0}^{C} \sum_{t \in \mathcal{Y}_c} \frac{\log(p_{tc})}{\sum_{c'=0}^{C} 1[t \in \mathcal{Y}_{c'}]} \tag{3.3}$$

$$\mathcal{L}_{video} = -\log(\tilde{p}_0) - \sum_{c=1}^{C} \frac{|\mathcal{Y}_c|}{|\mathcal{Y}^{pos}|} \log(\tilde{p}_c) \tag{3.4}$$

$$\mathcal{L}_{actionnness} = -\sum_{t \in \mathcal{Y}^{pos}} \log(a_t) - \sum_{t \in \mathcal{Y}_0} \log(1 - a_t) \tag{3.5}$$

Pseudo label mining

One additional implementation detail (only on the THUMOS [17] dataset, as per [29]) is positive pseudo label mining. This process is performed once after training the model for about half the total iterations. For each non-background class c, we start from each positive frame $t \in \mathcal{Y}_c$ and expand within the temporal radius r while the class probabilities p_{tc} remain high. Concretely, each step of the expansion increments a counter i, while $i < r$. Given a hyperparameter $\epsilon \in (0, 1)$, if $p_{t+i,c} > \epsilon p_{tc}$ we add $t + i$ to $\mathcal{Y}_c$; otherwise, the expansion process terminates. We then repeat in the backward direction with frames $t - i$.

Inference

Once the model has been trained, we follow the inference protocol of [29] without any hyperparameter tuning or multiple proposal generation with non-maximum suppression. Concretely, we compute the video-level score vector $\tilde{\mathbf{p}}$ and consider only the candidate class(es) $c = \{1, \dots C\}$ with $\tilde{p}_c > thr^v$ for a preset threshold thr^v. For each of these classes, we compute binary frame-level predictions $p'_{ct} = 1[(p_{ct} + \gamma a_t) > thr^f]$ (γ assigns the weight for actionness score and is set following [29]), binarized using another pre-set threshold thr^f, and generate contiguous segments (t_{start}, t_{end}) from these predictions. The confidence score associated with the segments are the max non-binarized frame score $(p_{ct} + \gamma a_t)$ within each segment.

3.3 Selective feature aggregation

The isolated features from each frame lack enough contextual expressive power for the task. Building upon the isolated features can often lead to noncontinuous predictions and lack of temporal structures or completeness. To remedy the limited representation ability on the temporal axis, existing methods heavily rely on the designs in

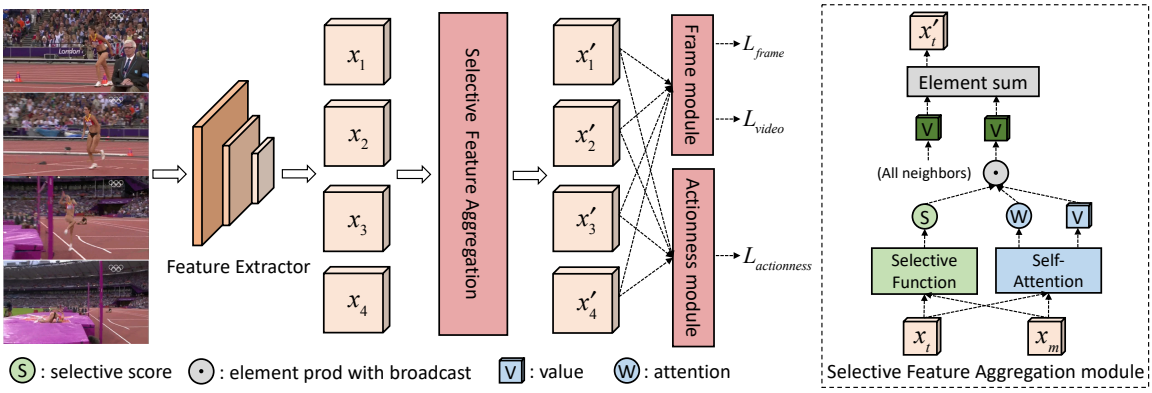

Figure 3.1: The schematic figure of our main model. Left side: we extract I3D frame features from a Kinetics-pretrained backbone [3]. The extracted isolated features are fed into the *Selective Feature Aggregation* module to obtain context-aware features. The selective functions in the module are designed to encode general inductive biases without accessing the labels for learning. For each frame, a frame module and an actionness module are used for class-aware and class-agnostic classification, respectively. Right side: details of the Selective Feature Aggregation module, which uses a selective function to further impose the prior through the attention scores used for feature aggregation.

processing the pseudo labels, such as refining the point-level prediction through a second phase module to find the consistent temporal chunk [18], or using complex optimal sequense search with outer-inner constrastive scoring and non-maximum suppression [22]. Although they work well, these post-processing methods can lead to arguably cumbersome systems.

To approach the temporal contextual information from an orthogonal angle, an obvious question to ask is: can we use feature aggregation for stronger temporal representations in the single frame-supervised setting? We thus propose the *Selective Feature Aggregation* module for the single frame setting, which performs a selective feature aggregation supported by both attention and a selective function along the temporal axis.

Given the feature set $\{\mathbf{x}_t\}_{t=1}^{T}$ for a video, we define a transformer-based aggregation module [43]. The module maps the feature to queries (Q), keys (K) and values (V), $Q, K, V \in \mathbb{R}^{T \times D'}$, where D' is the hidden dimension used in the trans-

former head. To aggregate features along the temporal axis, attention is computed to re-weight the values from each temporal location:

$$\mathbf{w}_t = \text{SOFTMAX}\left(\frac{\mathbf{q}_t K^\top}{\sqrt{D'}}\right), \mathbf{w}_t \in \mathbb{R}^T \tag{3.6}$$

and the refined per-frame feature is aggregated as $\mathbf{x}'_t = \sum_{m=1}^{T} w_{tm}\mathbf{v}_m$, where $\mathbf{q}_t, \mathbf{v}_m \in R^{D'}$ are the rows in Q and V respectively and w_{tm} is the m^{th} element of $\mathbf{w}_t$.

3.3.1 The selective functions

With the the scarcity of the annotations, a global attention-based feature aggregator will often lead to attention collapse or overfitting, since there is not enough information on the temporal locations. To combat those issues, a more general prior, which encodes the temporal inductive biases but doesn't rely on the labels, is needed. We introduce two selective functions to serve this purpose and help regularize the attention scores in the feature aggregation process.

Local window

A surprisingly effective selective function is a simple window-based step function with radius d constraining the receptive field: $s_d(t, m) = 1[|t - m| \leq d]$. The aggregated feature for frame t is then defined as $\mathbf{x}'_t = \sum_m s_d(t, m) w_{tm}\mathbf{v}_m$. This general prior encodes the information that local information has a higher chance to be relevant to frame t than others from a further location, and the information can be gathered in a parsimonious way. Note that we could also expand the local window methods to a multi-scale window, where we use multiple feature aggregators with different local window size to fuse the aggregated features.

Frozen feature similarity

Another source that provides the general information of similarities between frames is the frozen feature pre-trained on the large scale datasets. Without being impacted by the scarcity of the labels, the frozen feature similarities can directly be used to regularize the attention on the temporal axis. We calculate the cosine feature similarity: $\hat{s}(t,m) = \frac{x_t^T x_m}{x_t x_m}$ between two frames at timestamp t and m. To incorporate the similarity score into the attentions, we adopt two strategies: addition and product. For the addition operation, we add the cosine similarity score directly into the pre-softmax attention score: $\mathbf{w}_t = \textsc{softmax}\big(\frac{\mathbf{q}_t K^\top}{\sqrt{D'}} + \hat{s}(t)\big)$, where $\hat{s}(t) \in \mathbb{R}^T$ is the vector containing similarity scores $\hat{s}(t,m)$ between frame t and frames from all other locations. For the product operation, we first compute the score $s_p(t,m) = \text{sign}(\hat{s}(t,m))|\hat{s}(t,m)|^p$ with controllable sharpness using power p, then aggregate the contextual features as $\mathbf{x}'_t = \sum_m ((s_p(t,m)+1)/2) w_{tm} \mathbf{v}_m$.

We find that through using the simple selective functions, either the local window or the frozen feature similarity, the performance can be boosted significantly, especially on action-rich datasets where the scarcity of the location information has the most impact.

Chapter 4

Experiments

4.1 Setup

Datasets

We evaluate our proposed method on three challenging datasets. (1) THUMOS'14 [17] contains videos of 20 action classes with 200 validation and 213 test videos. This dataset is widely used for testing action detection models. We follow the common setup [29] and use the validation set for training and the test set for testing. (2) BEOID [6] is a dataset with 58 videos with 30 action classes in total. There are 12.5 action instances for each video on average. We follow the exact video train-test split set in [29], using 80% as training and 20% as testing. (3) GTEA [23] contains 28 videos of 7 fine-grained daily activities in a kitchen, where 21 videos are used for training and the rest for testing.

Evaluation metrics

We use the mean average precision (mAPs) under different intersection over union (IoU) thresholds to evaluate our model. A proposal instance is considered positive when both the action class is predicted correctly and the temporal IoU threshold

constraint is satisfied.

Implementation details

For all datasets, we use the Adam optimizer [19] with learning rate 1e-4. Aggregation modules tested in our experiment all use one round of feature aggregation. When using the self-attention based modules, the key and value embedding dimensions $D' = 2048$, for feed-forward networks we use 512 hidden units with a 0.3 dropout rate, and we use 8 heads with 0.3 dropout for the attention. For THUMOS'14, we train the model for 2000 iterations to obtain the pseudo labels, which are then used as label augmentation for another 3000 training iterations. For BEOID and GTEA, we train the model for 800 and 1500 iterations, respectively, without pseudo label mining. The background mining ratio η is 7 for THUMOS14 and BEOID; there is no background mining for GTEA. The actionness score weight γ in inference is 1.0.

4.2 Quantitative comparison with prior art

There are four key prior works tackling single-frame supervised action localization: **SF-Net** [29] is the framework described in Section 3.2, which uses the isolated frozen frame features as input to the entire training pipeline. This model provides a clean and basic testbed for single-frame supervised action detection. We use this model as our baseline, where our model only differs in the feature aggregation module. We rely on this model for the in-depth quantitative and qualitative analysis as it is the most directly comparable work.

The point-level **Ju et al.** [18] method utilizes both the frame-level and the proposal-level information to improve the action detection performance. Specifically, the method first generates frame-level key points, then further refines the key points using a second phase module to generate a proposal-level output. The model relies

Datasets	Methods	mAP@IoU							AVG	AVG
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.3-0.7	0.1-0.7
THUMOS'14 [17]	SF-Net [29]	71.0	63.4	53.2	40.7	29.3	18.4	9.6	30.2	40.8
	Ju et al. [18]	**72.8**	**64.9**	**58.1**	**46.4**	**34.5**	21.8	11.9	**34.5**	**44.3**
	Ours	71.0	64.6	56.5	45.3	33.8	**23.4**	**12.4**	34.3	43.9$^{+3.1}$
BEOID [6]	SF-Net [29]	62.9	-	40.6	-	16.7	-	3.5	-	30.1
	Ju et al. [18]	63.2	-	**46.8**	-	20.9	-	5.8	-	34.9
	Li et al. [24]	**71.5**	-	40.3	-	20.3	-	5.3	-	34.4
	Ours	67.1	-	40.9	-	**29.1**	-	**16.6**	-	**38.0**$^{+7.9}$
GTEA [23]	SF-Net [29]	58.0	-	37.9	-	19.3	-	11.9	-	31.0
	Ju et al. [18]	59.7	-	38.3	-	21.9	-	**18.1**	-	33.7
	Li et al. [24]	**60.2**	-	**44.7**	-	**28.8**	-	12.2	-	**36.4**
	Ours	57.2	-	39.7	-	22.8	-	17.9	-	33.8$^{+2.8}$

Table 4.1: Comparison with the state-of-the-art methods on three datasets (**best result**, second best). Our model shows competitive or higher performance across IOU metrics. Our module is simple, general and applicable for most frameworks. In the last column, we provide a direct comparison by showing the improvement over the baseline framework SF-Net [29] after plugging in our module.

on separately training a mapper to differentiably train the proposal component.

Similarly, the **Li et al.** [24] method works on the label space and proposes to use heuristics to detect action changes, with the help of an extra loss which enforces monotonicity of predictions to smooth out the inner region and build sharper boundaries. The method was tested on GTEA and BEOID datasets.

Finally, the very recent ICCV'21 work by **Lee et al.** [22] obtains impressive results but utilizes strong dataset-specific priors for frame mining. Note that we adopt the more general frame mining framework proposed in [29], so the results are not directly comparable.

The comparison between our method and prior works is summarized in table 4.1. For each dataset, we report the best numbers obtained through our model variants: the multi-scale window feature aggregator (with window sizes 10, 30, and 60) for THUMOS'14, the local window with size 60 for BEOID and the frozen-feature similarity with the addition operation for GTEA. On all the datasets, we obtain competitive or higher performance on average IoUs. Specifically, on THUMOS'14, we achieve mAP 34.3% on IoU averaged between 0.3-0.7 and 43.9% on IoU averaged between 0.1-0.7. Compared with our baseline method SF-Net [29], we show that simply through adding

	THUMOS'14					BEOID					GTEA				
Metric	mAP@IoU				AVG	mAP@IoU				AVG	mAP@IoU				AVG
	0.1	0.3	0.5	0.7		0.1	0.3	0.5	0.7		0.1	0.3	0.5	0.7	
global	70.0	55.1	32.0	10.6	42.4	68.1	40.0	24.0	13.1	35.0	51.2	33.5	20.7	16.5	29.9
local-small	70.1	55.7	33.6	12.2	43.2	67.8	39.3	30.4	13.6	36.5	56.6	38.5	23.0	**19.0**	33.4
local-med	**70.6**	**56.3**	32.8	11.8	**43.4**	67.1	41.0	29.1	**16.6**	**38.0**	57.2	**39.7**	22.8	17.9	33.6
frozen-prod	68.6	54.5	**33.7**	**12.6**	43.0	70.9	40.7	25.5	12.6	36.4	**58.5**	38.0	22.2	16.3	33.2
frozen-add	68.0	54.6	32.9	**12.6**	42.4	**71.4**	36.0	23.7	10.0	34.5	56.8	38.8	**24.3**	17.1	**33.8**

Table 4.2: The ablation studies on various designs of selective functions on THUMOS'14, BEOID and GTEA. The AVG score is computed across all IoUs from 0.1 to 0.7. For the global feature aggregation, we use 700, 200, and 60 for THUMOS'14, BEOID, and GTEA, respectively. Two variants of local windows are tested. The local-small indicates a compact range and is set as 5, 7, and 5; the local-medium attempts to expand the small range and is set as 30, 60 and 20, respectively for the datasets from left to right.

our proposed simple feature aggregation module, the performance is boosted 3.1% on THUMOS'14 (from 40.8% to 43.9% mAP over IoUs 0.1-0.7), 7.9% on BEOID (from 30.1% to 38.0%) and 2.8% on GTEA (from 31.0% to 33.8%). Especially, on the challenging localization setting of IoU 0.7, our method outperforms the strong recent model of Ju et al. [18] by 2.8% on THUMOS'14 and 10.8% on BEOID respectively, and is only 0.2% behind on GTEA despite not using a second-phase refinement system on the predictions.

4.3 Ablation studies

We conduct ablation studies on the two key components of our proposed method: the contextual features and the selective functions.

Contextual features

In table 4.3 we compare the baseline model, SF-Net [29] (which uses the isolated features for each frame), the mean-pooled features (which simply averages the frame features within a local window rather than using the learning attention weights), and our selective feature aggregation module. We keep window sizes the same for mean-pool and selective attention for a fair comparison. We find it interesting that,

Feature-agg	mAP@IoU				AVG
	0.1	0.3	0.5	0.7	
THUMOS'14					
isolated	**71.0**	53.2	29.3	9.6	40.8
mean-pool	66.8	53.2	32.7	12.0	41.6
selective+attn	**71.0**	**56.5**	**33.8**	**12.4**	**43.9**
BEOID					
isolated	62.9	40.6	16.7	3.5	30.1
mean-pool	63.4	32.9	21.4	11.3	30.9
selective+attn	**67.1**	**40.9**	**29.1**	**16.6**	**38.0**
GTEA					
isolated	**58.0**	37.9	19.3	11.9	31.0
mean-pool	47.62	28.5	11.15	7.6	23.2
selective+attn	57.2	**39.7**	**22.8**	**17.9**	**33.8**

Table 4.3: We compare: the isolated feature, mean-pooled feature, and selective feature aggregation. All variants are based-on the SF-Net framework, and the isolated feature corresponds to the SF-Net [29] baseline.

on THUMOS'14 and BEOID, the mean-pooled feature without location-specific information can already improve the results by 0.8%, indicating the importance of the contextual information. With our proposed module with selective functions and dynamic attention, the results can be improved by 3.1%, 7.9% and 2.8% over isolated features, and 2.3%, 7.1% and 10.6% over the mean-pooled features on the three datasets.

Selective functions

We show that having an effective general prior to perform selective neighbor feature aggregations can further improve the performance of our model. Summarized in table 4.2, we compare our selective function designs under all three datasets.

We first analyze our intuition that the local information tends to have a higher chance to be relevant compared to far-away frames. As a comparison, we extend the window size to global ones which have much larger aggregation receptive fields. For local window sizes, we consider small and medium-size windows to analyze the impact of selective functions. We find that the performance drops by 1.0%, 3.0%, and 3.7% when a global-size window is used. It's also interesting that, for the local window

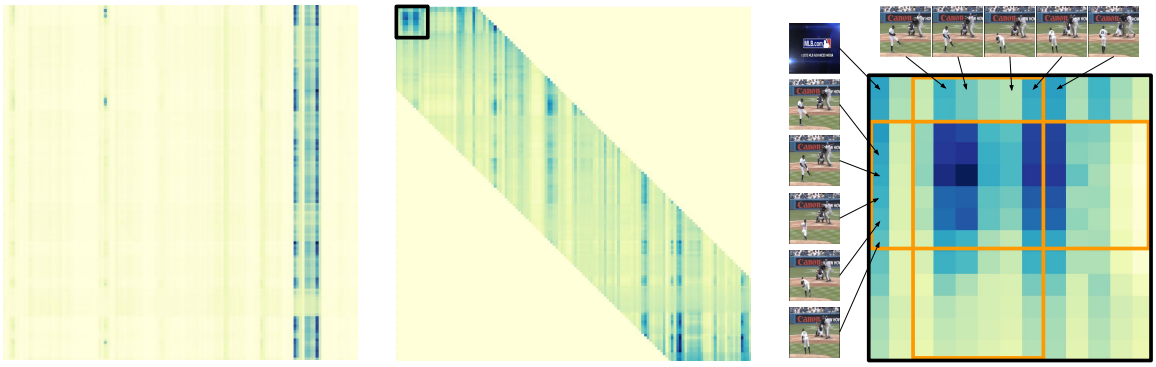

Figure 4.1: Attention matrix visualization between video frames (every row corresponds to the attention vector $\mathbf{w}_t$ computed for frame t over all other video frames). Figures from left to right: global attention matrix, local attention matrix with window size 30, and zoomed-in attention matrix of 1^{st} to 13^{th} frame from a "BaseballPitch" video in THUMOS14. Darker color indicates larger attention weights. The black box indicates the zoomed-in region. The orange rectangles show the ground truth action frames.

selective functions, even with a small window size, such as 5 or 7, the model can still outperform the global-window aggregation, indicating that learning on the necessary information has a less chance to lead to overfitting on the training frames.

For the frozen feature similarity selective functions, we compare the results from the addition function and the product function. We find that, similar to local windows, the frozen features can also provide a general prior to regularize the feature aggregation and lead to a performance boost, compared to either the isolated features [29] or the global feature aggregators. Results are summarize din the last two rows of table 4.2.

4.4 Qualitative Analysis

In this section, we provide additional qualitative analysis on the model's behavior.

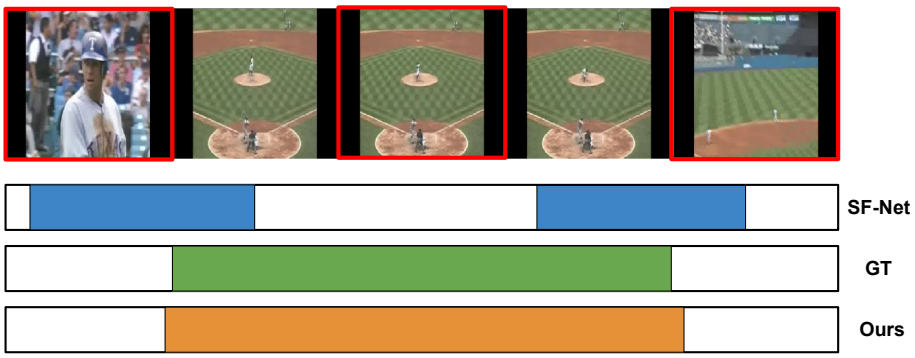

Figure 4.2: Qualitative comparison with SF-Net on THUMOS'14. The red box around the video frame indicates either false positive or false negative frames misclassified by SF-Net but is correctly predicted by our method. The result from our method is more accurate and contiguous on the temporal axis.

Collapsed attention

We visualize the attention scores learned from a global feature aggregation module and a local window-constrained feature aggregation module in figure 4.1. With global attention module, we find that only a small set of frames (and always the same frames) are selected for feature aggregation, potentially due to that the target labels only contain very sparse information. If we constrain the model's attention using the simple local window prior, the model can learn to exploit more diverse features from various locations for prediction, preventing the model from overfitting on a small set of frames. We also visualize the zoomed-in attention scores with the key frames in figure 4.1, showing that our model attends to the relevant keys frames for feature aggregations.

Prediction results analysis

We visualize the prediction results of our model with the Selective Feature Aggregation module, and find that our model tends to make predictions with more temporal consistency, compared to baseline [29], shown in figure 4.2. We also find that our method tends to produce more accurate predictions under the same recall, for ex-

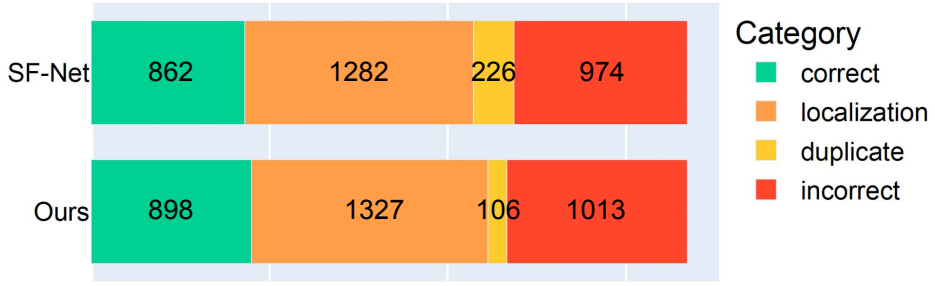

Figure 4.3: For every class, we take the N highest-scoring detections returned by our model and SF-Net [29], where N is the number of ground truth instances for this class. We perform error analysis by assigning each detection to one of four categories: (1) *correct*, where the detection matches a ground truth action instance with IoU>0.7 and has a higher score than any other competing detection; (2) *localization*, where the detection matches a ground truth action instance but only with IoU>0.3; (3) *duplicate*, where the detection matches a ground truth instance with IoU>0.3 but there is already a higher-scoring detection matched to this action instance; and (4) *incorrect*, which is all other detections. We note that we have cut the number of duplicate detections in half compared to SF-Net, which is consistent with our intuition in figure 4.2.

ample, on video-level classification, our method improves 2.5% (83.5% vs. 81%) in precision while maintaining the same recall (98%) as SF-Net. We analyze the predicted detections in figure 4.3, and find that our method has less duplicate predictions, indicating that more contiguous predictions are generated, which is consistent with our observation in figure 4.2.

4.5 Model performance under the same budget

Annotation costs are the key bottleneck of video-related applications. To demonstrate the cost-effectiveness of our method, we compare it with the state-of-the-art models under the same annotation budgets. According to Ma et al. [29], the labeling time required for video-level, single-frame, and full annotations on a single GTEA video is 45, 50, and 300 seconds. Following the estimate of the time budgets, we instead benchmark all the models on the THUMOS'14 dataset, which is much larger than GTEA and BEOID.

Specifically, we need 200, 180, and 30 videos for training for weakly-supervised, single-frame supervised, and fully-supervised methods. In the fully supervised case, we increase the number of videos to 42 to ensure that there are at least 2 videos per action class. Similarly, for the single-frame supervised setting, to conduct a fair comparison and maintain the class-wise distribution, we removed one video from each class to obtain the 180 videos for training. We use the original test video set for testing.

The results are summarized in table 4.4, we find that our method, under the same fixed budget for annotation, achieves the best performance compared to the previous fully-supervised, video-level supervised, and single-frame supervised methods, achieving 42.9% on the test set.

Supervision	#videos	Method	AVG mAP
Full	42	PGCN*[50]	39.5
Single frame	180	SF-Net[29]	40.2
Video-level	200	CoLA[51]	40.9
Single frame	180	**Ours**	**42.9**

Table 4.4: AVG(0.1-0.7) mAP on THUMOS14 under a fixed human labeling budget; #videos indicates the corresponding number of training videos. *Note the PGCN method uses class-agnostic temporal proposals that have been trained using the full training set.

Chapter 5

Conclusion

In this thesis, we presents a new Selective Feature Aggregation module in the single frame-supervised setting, which dynamically computes the attention over frames under the guidance of a selective function to produce contextualized features for each frame. We find that the contextualized features can improve the performance over the baseline by a large margin, and that the selective functions can effectively regularize the attention to avoid the collapse issue. The proposed module is simple, effective and applicable to various single frame-supervised action detection frameworks.

Appendix A

Video Sequence Sorting

A.1 Video Sequence Sorting

Aggregation & Task	Frame Accuracy	Video Accuracy
isolated & binary	67.3	68.7
selective & binary	$68.6^{+}1.3$	$70.3^{+1.6}$
isolated & six-way	16.5	16.9
selective & six-way	$17.4^{+0.9}$	$16.6^{-0.3}$

Table A.1: Frame-level accuracy and video-level accuracy comparison between baseline isolated features and selectively aggregated features, on two-frame sorting task, binary classification, and three-frame sorting task, six-way classification.

Task Overview

Another motivation for our selective feature aggregation method comes from our observation that frozen extracted features perform poorly on the video sequence sorting task. The frozen features, extracted from pretrained network on task of video recognition, lack fined-grained spatio-temporal details to perform well on video sequence sorting task. The video sequence sorting task requires the model to output the correct video sequence order given a randomly shuffled video sequence [21]. Features need to contain enough fine-grained spatial-temporal details in order to correctly sort the

video sequence.

Model

We follow the model architecture of Order Prediction Network [21] and experiment on two classification tasks: sorting two-frame sequences, a binary (2!) classification task, and sorting three-frame (3!) sequences, a six-way classification task.

Train Data

We prepare the training data from THUMOS14. For each video in training dataset and for each frame that has single-frame label, we randomly shuffle the video frame sequence around that labeled frame. For two-frame sorting task, we take the labeled frame and one of its neighboring frame. For three-frame sorting task, we take the labeled frame, one backward neighboring frame and one forward neighboring frame. The reason for including the labeled frame in every shuffled sample is to make sure that every sample contains at least one discriminative action frame. We also take stride with size two when taking the neighboring frames around the labeled frame, in order to prevent the frames in the sequence from being too similar to each other. Intuitively, if the frames in the sequence are indistinguishable from each other, it will be very hard to sort them.

Experiments

We train Order Prediction Network [21] on 150 training videos and report frame-level accuracy and video-level accuracy on 50 testing videos. The frame-level accuracy is percentage of correctly predicted sequence of all testing sequences. The video-level accuracy is computed by first calculating the frame-level accuracy of each test video and then taking the average across all videos. Table A.1 presents the experiment results.

Neither isolated or aggregated features on three-frame sorting task are much better than random guess (16.6%), which may suggest the lack of fine-grained details of the frozen extracted features. Our selectively aggregated features generally improves upon the baseline isolated features on both two-frame and three-frame sorting tasks. However, the improvement is not very significant. We hypothesize that selectively aggregating features does help but aggregating already coarse features does not achieve the level of spatio-temporal granularity required by the sorting tasks.

Bibliography

[1] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[2] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.

[3] J. Carreira and A. 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.

[4] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. 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.

[5] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017.

[6] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, volume 2, page 3, 2014.

[7] N. Ding, X. Fan, Z. Lan, D. Schuurmans, and R. Soricut. Attention that does not explain away. *arXiv preprint arXiv:2009.14308*, 2020.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[9] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013.

[11] J. Gao, Z. Yang, K. Chen, C. Sun, and R. 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.

[12] F. C. Heilbron, W. Barrios, V. Escorcia, and B. Ghanem. Scc: Semantic context cascade for efficient action detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3175–3184. IEEE, 2017.

[13] F. C. Heilbron, J. C. Niebles, and B. 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.

[14] A. Islam and R. Radke. Weakly supervised temporal action localization using deep metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 547–556, 2020.

[15] A. Jabri, A. Owens, and A. A. Efros. Space-time correspondence as a contrastive random walk. *Advances in Neural Information Processing Systems*, 2020.

[16] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 740–747, 2014.

[17] Y.-G. 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. http://crcv.ucf.edu/THUMOS14/, 2014.

[18] C. Ju, P. Zhao, S. Chen, Y. Zhang, Y. Wang, and Q. Tian. Divide and conquer for single-frame temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13455–13464, 2021.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[20] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[21] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 667–676, 2017.

[22] P. Lee and H. Byun. Learning action completeness from points for weakly-supervised temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13648–13657, 2021.

[23] P. Lei and S. Todorovic. Temporal deformable residual networks for action segmentation in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6742–6751, 2018.

[24] Z. Li, Y. Abu Farha, and J. Gall. Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2021.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *International Conference on Computer Vision (ICCV)*, 2021.

[27] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[28] Z. Luo, D. Guillory, B. Shi, W. Ke, F. Wan, T. Darrell, and H. Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *European conference on computer vision*, pages 729–745. Springer, 2020.

[29] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, and Z. Shou. Sf-net: Single-frame supervision for temporal action localization. In *European conference on computer vision*, pages 420–437. Springer, 2020.

[30] D. Moltisanti, S. Fidler, and D. Damen. Action recognition from single timestamp supervision in untrimmed videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9915–9924, 2019.

[31] A. Montes, A. Salvador, S. Pascual, and X. Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. In *1st NIPS Workshop on Large Scale Computer Vision Systems*, December 2016.

[32] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8679–8687, 2019.

[33] M. Nawhal and G. Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021.

[34] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.

[35] D. Oneata, J. Verbeek, and C. 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.

[36] S. Paul, S. Roy, and A. 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.

[37] M. Rashid, H. Kjellstrom, and Y. J. Lee. Action graphs: Weakly-supervised action localization with graph convolution networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–624, 2020.

[38] B. Shi, Q. Dai, Y. Mu, and J. Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1019, 2020.

[39] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017.

[40] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018.

[41] Z. Shou, D. Wang, and S.-F. 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.

[42] K. Tang, B. Yao, L. Fei-Fei, and D. Koller. Combining the right features for complex event recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2696–2703, 2013.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[44] L. Wang, Y. Xiong, D. Lin, and L. 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.

[45] X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[46] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[47] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[48] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[49] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2016.

[50] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019.

[51] C. Zhang, M. Cao, D. Yang, J. Chen, and Y. Zou. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16010–16019, June 2021.

[52] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[53] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.

[54] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.