Deep Motion Prior for Weakly-Supervised Temporal Action Localization

Meng Cao, Can Zhang, Long Chen, Mike Zheng Shou, Member, IEEE, and Yuexian Zou

Abstract—Weakly-Supervised Temporal Action Localization (WSTAL) aims to localize actions in untrimmed videos with only video-level labels. Currently, most state-of-the-art WSTAL methods follow a Multi-Instance Learning (MIL) pipeline: producing snippet-level predictions first and then aggregating to the video-level prediction. However, we argue that existing methods have overlooked two important drawbacks: 1) inadequate use of motion information and 2) the incompatibility of prevailing cross-entropy training loss. In this paper, we analyze that the motion cues behind the optical flow features are complementary informative. Inspired by this, we propose to build a context-dependent motion prior, termed as motionness. Specifically, a motion graph is introduced to model motionness based on the local motion carrier (e.g., optical flow). In addition, to highlight more informative video snippets, a motion-guided loss is proposed to modulate the network training condition on the motionness scores. Extensive ablation studies confirm that motionness efficaciously models action-of-interest, and the motion-guided loss leads to more accurate results. Besides, our motion-guided loss is a plug-and-play loss function and is applicable with existing WSTAL methods. Without loss of generality, based on the standard MIL pipeline, our method achieves new state-of-the-art performance on three challenging benchmarks, including THUMOS’14, ActivityNet v1.2 and v1.3.

Index Terms—Weakly-supervised temporal action localization (WSTAL), deep motion prior, motion-guided loss.

I. INTRODUCTION

TEMPORAL Action Localization (TAL) aims at identifying the start and end timestamps of all the action instances occurring in an untrimmed video. It is an indispensable building block for numerous video understanding applications such as intelligent video summarization [9], surveillance analysis [26], and video retrieval [3], etc. However, the required frame-level labeling is subjective, labor-intensive, and error-prone. Therefore, Weakly-Supervised Temporal Action Localization (WSTAL) has gained intensive attention, where the “weak” denotes a much cheaper labeling cost at the video level. For example, as shown in Fig. 1, given the untrimmed video and category Billiards, WSTAL models aim to detect the temporal locations of all Billiards action instances. Unfortunately, there is no free lunch — it is essentially challenging to perform frame-wise localization and classification with only the video-level supervision, especially for complex visual scenes.

The de facto paradigm of most state-of-the-art WSTAL methods [7], [14], [15], [17], [28] is to first pre-extract video RGB and optical flow features from pre-trained vision models (e.g., C3D [25] and I3D [2]), then follow the Multi-Instance Learning (MIL) principle. Specifically, snippet-wise classifications are performed over time to generate Temporal Class Activation Sequence (TCAS), which is then aggregated to predict the video-level classification score. The whole process is optimized by a standard cross-entropy (XE) loss. We call this framework baseline in our paper.

Although great progress has been made in this direction, two main drawbacks still exist in today’s WSTAL methods: (i) Inadequate use of motion information: We argue that the motion cue behind the optical flow modality is empirically more informative (cf. Sec. IV-D) and should not be treated equally as the RGB modality. Most existing methods, however, fail to extensively explore the motion information behind optical flow. Specifically, current WSTAL methods always resort to either early fusion or late fusion manner. For early fusion methods [11], [17], [21], the extracted RGB and optical flow features are concatenated before feeding to
the network. For late fusion methods [15], [36], they conduct predictions on each modality independently and then fuse the results by the weighted sum or other post-processing steps. However, this simple concatenation or fusion manners are indirect and inadequate, leading to many false detection results. For example, as shown in Fig. 1, frame #1 shares similar appearances with frame #2, but it is actually a background frame since it is only a stationary frame with explanatory narrations. In this case, the motion information can help us to easily distinguish them because their optical flow features have low responses, which implies that this frame is unlikely to be an action. (ii) Incompatibility of XE loss: XE loss is designed to measure the performance of a classification model and is inherently incompatible with our localization task. Specifically, the XE loss encourages the discriminative video clips that tend to be fragmentary without covering the entire actions [10], [24], [33], [40]. For example, frame #3 is within a ground truth interval, yet it is misclassified and leads to incomplete localization results. Similarly, let’s focus on the motion information. From the optical flow images, we find similar patterns between frame #3 and frame #2. Thus, introducing coherent motion information in the loss function will lead to more complete and accurate predictions.

Motivated by this, we propose Deep Motion Prior Network (DMP-Net) to make full use of the optical flow modality via learning an effective context-dependent motion representation (referred to as motionness in our paper). Our motionness is with global perception and focuses on action-of-interest regardless of the background and irrelevant motions. Based on this, we propose a motion-guided loss, which is a plug-and-play loss function that may be an alternative to the traditional XE loss under the weakly-supervised setting.

For the motionness modeling, we introduce a motion graph to enlarge the receptive field of each temporal snippet since optical flow is the local motion representation calculated between two consecutive frames. Specifically, this paper investigates the temporal relationships from two perspectives: positional relationship and semantic relationship. To illustrate this, let’s revisit frame #3 in Fig. 1. 1) positional relationship: frame #3 is misclassified while its surrounding frames are all correctly predicted. These adjacent snippets will provide contextual information, which is advantageous for correct classification. 2) semantic relationship: frame #2 is distant from frame #3, but they share the similar motion patterns (similar billiard ball hitting processes). Thus, it provides indicative hints for frame #3 and leads to more comprehensive information. Based upon these two relationships, we construct the motion graph which encourages both the adjacent positional contexts and the distant semantic correlations.

For motion-guided loss, we aim to use the modeled motionness to modulate the network training. We also start from an intuitive idea that the higher motionness of one timestamp, the greater the probability of it becoming action-of-interest. Let’s recall the video-level classification aggregation process in the baseline. Following the top-k mean strategy in [7], [16], [17], [37], for each category, the mean value of k terms with the largest TCAS values is computed as the video-level classification score. In this paper, we further evaluate the motion characteristics of these selected terms. Specifically, we take the values of the corresponding terms in the motionness sequence and incorporate these values into the loss computation. In this way, terms with both high TCAS and motionness scores are highlighted while terms with low motionness scores are down-weighted. Experimental results have shown that this intuitive design leads to better results.

In summary, we make three contributions in this paper:

- We argue that a context-dependent deep motion prior is crucial in accurate action localization and we obtain it by applying a motion graph to exploit the relationships between temporal nodes.
- An efficient motion-guided loss is developed to inform the whole pipeline of more motion cues, which can be seamlessly plugged into any existing WSTAL models.
- Extensive experiments on three challenging datasets have demonstrated the effectiveness of our proposed DMP-Net.

II. RELATED WORK

A. Weakly-Supervised Temporal Action Localization

WSTAL requires only video-level labels and saves large human effort from the frame-level labeling. The primer work UntrimmedNets [28] formulates this problem as a Multi-Instance Learning (MIL) framework by evaluating the contribution of each clip to the video classification. Later, STPN [15] applies this attention mechanism to the feature level with a proposed sparsity constraint. To regularize the feature representation, W-TALC [17] introduces deep metric learning as a complement. Inherently, attention-MIL methods tend to produce incomplete localization results. To tackle this, several works [10], [24], [34] try to extend the discriminative regions via randomly hiding patches or suppressing the dominant response. To model complete actions, Liu et al. [10] develop a parallel multi-branch classification architecture with the help of the generated hard negative data. Zeng et al. [34] proposes an iterative training strategy, which selects the most discriminative action instances in each iteration and removes them in the next iteration. For the clear distinction between background and foreground, several background modeling works [7], [16], [20] are proposed to ease the action-context confusion. CleanNet [11] introduces additional pseudo-supervision by leveraging the temporal contrast in snippet-level action classification predictions. UM-Net [8] models the background frames as out-of-distribution samples and realizes this uncertainty learning via multiple instance learning. SF-Net [12] introduces extra supervisions by annotating one single frame within the action and mines pseudo action and background frames based on this annotation.

Typically, most WSTAL methods are based on the extracted RGB and optical flow with two possible fusion ways. Early fusion methods [7], [11], [17], [21] concatenate two modalities before feeding to the network while late fusion methods [15], [16] compute a weighted sum of their respective outputs. Either of the two fusion strategies treats RGB and optical flow equally or independently. However, the motion cue behind the optical flow modality is empirically more informative. Experimental results in previous publications [36] and our experiments (Sec. IV-D) have also demonstrated that when using the single modality, optical flow based methods are
and (bottom plot) Guidance branch to output class-agnostic deep motion prior. In the base branch, for each channel (category) of TCAS, top-
motionness sequence is also selected and fed to our Motion-guided Loss
and long-term semantic information. In contrast, our sparse graph with positional edges
in videos, making the learned weight focus on surrounding
fully connected way suffers from the feature slowness [38]
node pair is connected. In our experiments, we find that this
i
features while our DMP-Net focuses on the effective motion
is applied on the concatenation of the RGB and optical flow
following two aspects. Firstly, in [19], the graph convolutional network on the extracted region features.
proposals to refine the boundary regression. G-TAD [30]
incorporates multi-level video context into feature represen-
tion (termed as motionness) based on the optical flow
branch, we seek to build a context-dependent motion repre-
sentation. We try to emphasize the importance of the optical flow
modality over the RGB one and solve the incompatibility problem of cross-entropy loss. Thus, we propose a gen-
eral motion-guided loss that may replace the traditional cross-entropy loss in the existing WSTAL methods in a plug-
and-play manner. To demonstrate the effectiveness of the proposed loss, we design an effective WSTAL model called
DMP-Net, which consists of two branches: the base branch and the guidance branch. The whole network is optimized with our
motion-guided loss. The overall pipeline of our DMP-Net is demonstrated in Fig. 2.

B. Overview of DMP-Net

We try to emphasize the importance of the optical flow
modality over the RGB one and solve the incompatibility problem of cross-entropy loss. Thus, we propose a gen-
eral motion-guided loss that may replace the traditional cross-entropy loss in the existing WSTAL methods in a plug-
and-play manner. To demonstrate the effectiveness of the proposed loss, we design an effective WSTAL model called
DMP-Net, which consists of two branches: the base branch and the guidance branch. The whole network is optimized with our
motion-guided loss. The overall pipeline of our DMP-Net is demonstrated in Fig. 2.

Without loss of generality, we take the standard multi-instance learning pipeline as the base branch. Specifically, the RGB and optical flow features are concatenated to generate snippet-level classification results, i.e., Temporal Class Activation Sequence (TCAS) [17]. For the guidance branch, we seek to build a context-dependent motion representation (termed as motionness) based on the optical flow features. To this end, a graph convolutional module is intro-
duced to model snippet-level relations. Instead of directly con-
structing a dense fully-connected graph, we carefully design a computa-
tion-efficient graph with sparse edge connections. Here we take two types of relations, i.e., positional edges
to utilize neighborhood correlations and semantic edges to capture semantically related but disjointed snippets. Finally,
the generated motionness is used to guide the network training in our motion-guided loss.

C. Base Branch

The base branch is shown in the top part of Fig. 2. Firstly, we apply an embedding function $f_{\text{embed}}$ over the concatenation
of $X_n^{Apr}$ and $X_n^{Mot}$ to obtain the features $X_n^E \in \mathbb{R}^{T \times 2d}$. $f_{emb}$ is implemented with a temporal convolution followed by the ReLU activation function. Given the embedded features $X_n^E$, we apply a classifier $f_{cls}$ to obtain snippet-level class scores, namely Temporal Class Activation Sequence.

$$A_n = f_{cls}(X_n^E; \phi_{cls}),$$

(1)

where $f_{cls}$ contains a temporal convolution followed by ReLU activation while $\phi_{cls}$ are the learnable parameters. The obtained $A_n \in \mathbb{R}^{T \times C}$ represents the action classification results occurring at each temporal snippet.

**D. Guidance Branch**

In this branch, we aim to model a context-dependent motion prior called motionness. Traditionally, the optical flow feature is widely adopted to provide temporal motion information [19], [23], [36]. However, optical flow has an inherent disadvantage that it can only reflect the local motion information since it is computed between two consecutive frames. To obtain an effective motion prior, we build a motion graph to model the snippet-snippet relations and thus eliminate interfering motion information (e.g., background or unrelated motions). Thus, we present our motion graph construction process and motionness modeling with GCNs sequentially.

1) **Motion Graph Construction:** For the graph construction, one possible way is to build a dense fully connected graph, i.e., each snippet is connected with the other snippets across the whole video. This intuitive way suffers from the two drawbacks: (i) The dense connection way is expensive and the number of edges is the quadratic order of the number of snippets; (ii) Since the short-term video varies slowly, this slowness prior [38] leads to the large feature similarity between adjacent snippets. Thus the feature updating within the dense graph is dominated by surrounding snippets because of the similar feature representations and the faraway snippets are ignored.

To alleviate this, we build a motion graph with sparse edge connections and encourage both the positionally adjacent snippets and the semantically related but disjointed snippets. Formally, let $\mathcal{G} = \{\mathcal{M}, \mathcal{E}\}$ be the graph of $T$ nodes with the node set $\mathcal{M}$ and edge set $\mathcal{E}$. Furthermore, the adjacency matrix associated with $\mathcal{G}$ is denoted as $G \in \mathbb{R}^{T \times T}$. For the graph $\mathcal{G}$, each node (i.e., $m_i \in \mathbb{R}^d$) is instantiated as the optical flow features of the corresponding snippet. Here, we devise two types of edge construction approaches by exploiting both the positional awareness and the semantic similarities, i.e., $\mathcal{E} = \mathcal{E}_{pos} \cup \mathcal{E}_{smt}$.

2) **Positional Edges $\mathcal{E}_{pos}$:** Snippets that are close in location have a natural temporal connection and the message passing among them will facilitate the feature representation. As the example shown in Fig. 3, s1, s2, and s3 are the different stages of the GolfSwing action. Thus s1 and s3 provide fruitful contextual information for the modeling of s2. Generally, we establish an edge between snippet $m_i$ and $m_j$ if their temporal distance is below one pre-set threshold:

$$\mathcal{E}_{pos} = \left\{ e_{ij} \mid \left| \frac{t_i - t_j}{T} \right| < \theta_{pos} \right\},$$

(2)

where $e_{ij} = (m_i, m_j)$ is the edge connecting $m_i$ and $m_j$; $t_i$ and $t_j$ are the temporal indices of snippet $m_i$ and $m_j$, respectively. $\theta_{pos}$ is a certain threshold. In this way, each snippet is enhanced with its contextual information, which is obviously helpful to refine the feature representation and increase the localization accuracy.

3) **Semantic Edges $\mathcal{E}_{smt}$:** Besides the temporal context information, the semantic correlation is also beneficial for each snippet even when they are scattered in time. Note that the untrimmed video often contains multiple action segments. Thus, finding the action instances belonging to the same or similar categories will enhance the discriminative motion patterns. As shown in Fig. 3, s4 and s5 are both the golf swing actions across the different scenes and actors. Despite the long temporal distance, s4 and s5 share similar semantic information with s2, and this kind of connection is beneficial for the motionness modeling. Thus we set the semantic edges for long-range correlation modeling, which helps build high-level and global-aware relations for the same or similar action instances within one video. In particular, we use the cosine similarity between snippet nodes to find semantically related nodes, i.e.,

$$\mathcal{E}_{smt} = \left\{ e_{ij} \mid \frac{|W_{m_i}^\top W_{m_j}|}{\|W_{m_i}\|_2 \cdot \|W_{m_j}\|_2} > \gamma \right\},$$

(3)

where $\gamma$ is the similarity threshold. $W_1, W_2 \in \mathbb{R}^{d \times d}$ are learnable parameters.

4) **Motionness Modeling:** Given the constructed graph, we apply $K$-layer GCNs on the constructed motion graph to perform reasoning. GCNs facilitate the message passing of the graph and update motion features for each snippet node. Concretely, for the $k$-th layer:

$$X^k = GX^{k-1}W^k, 1 \leq k \leq K,$$

(4)

where $W^k \in \mathbb{R}^{d_k \times d_k}$ is the learnable parameter matrix; $X^k \in \mathbb{R}^{T \times d_k}$ are the hidden motion features for all snippets at layer $k$; $X^0 = \{m_i\}_{i=1}^T \in \mathbb{R}^{T \times d}$ are the input motion features. $G$ is the adjoint matrix formulated as follows:

$$G_{ij} = \begin{cases} m_i^\top m_j \|m_i\|_2 \|m_j\|_2, & e_{ij} \in \mathcal{E}; \\ 0, & e_{ij} \notin \mathcal{E}. \end{cases}$$

(5)
Note that following the common practice [30], [35], a shortcut path is used to preserve the input features and the final output is as follows:

\[ X^K = X^K \| X^0, \] (6)

where \( \| \) denotes the concatenation operation. The graph convolutions allow the network to compute the response of a node based on its edges defined by the graph, thereby enlarging the receptive field and facilitating information exchange among neighboring or distant snippets.

After obtaining the updated motion features \( X^K \), we apply a binary classifier \( f_{mot} \) to obtain the motionness. Specifically, the classifier contains a temporal convolution followed by the ReLU activation function. This can be formulated as follows:

\[ M_a = f_{mot}(X^K; \phi_{mot}), \] (7)

where \( \phi_{mot} \) is the learnable parameter. The obtained \( M_a \in \mathbb{R}^T \) represents the motionness scores for each temporal snippet.

### E. Motion-Guided Loss

Before we specify our motion-guided loss, let’s revisit the commonly-adopted video-level classification loss, which is in a traditional binary cross-entropy form.

To get the video-level class scores, we aggregate snippet-level class scores computed in Eq. (1). Following [7], [17], [28], we take the top-\( k \) mean strategy: for each class \( c \), we take \( \{L_n^r\} \) \( \{r\} \) terms with the largest class-specific TCAS values and compute their means as \( a_{n,c} \), namely the video-level class score for class \( c \) of video \( V_n \). The index set of the corresponding selected elements is denoted as \( S_{n,c}^r \). After obtaining \( a_{n,c} \) for all the \( C \) classes, we apply Softmax function on \( a_n \) along the class dimension to get the video-level class possibilities \( p_n \in \mathbb{R}^C \), namely \( p_n = \text{softmax}(a_n) \). XE loss (\( \mathcal{L}_a \)) is then calculated in the cross-entropy form:

\[ \mathcal{L}_a = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \hat{y}_{n,c} \log(p_{n,c}), \] (8)

where \( \hat{y}_{n} \in \mathbb{R}^C \) is the normalized ground-truth.

XE loss is only associated with the TCAS sequence, which is modeled based on the motion and appearance feature concatenation. In previous discussions, we have noticed that motion information provides more hints to facilitate the localization. Thus the commonly-adopted top-\( k \) mean strategy is a rather coarse aggregation manner since it does not consider the motionness score for each selected term. To alleviate this, we additionally generate the video-level motionness scores \( \mu_n = \{\mu_{n,c}\} \in \mathbb{R}^C \) as the mean motionness value for those snippets with top-\( k \) motionness values:

\[ \mu_{n,c} = \text{mean}(M_{n,r} | r \in S_{n,c}^r), \] (9)

where \( S_{n,c}^r \) is the index set of the top-\( k \) terms for category \( c \) in the TCAS sequence as previously defined.

Intuitively, the network should highlight snippets that are simultaneously correctly classified and have a high motionness score, i.e., the motion-guided loss is:

\[ \mathcal{L}_g = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \mu_{n,c}^2 \log(p_{n,c}) - \log \mu_{n,c}^2, \] (10)

where \( \mu_{n,c} \) is the same as previously defined in Eq. (8).

We interpret this formulation of Eq. (10) term by term intuitively. The first term aims to highlight the snippets with both high \( p_{n,c} \) and high motionness (\( \mu_{n,c} \)) while the second term (\( -\log \mu_{n,c}^2 \)) serves as a regularization term to prevent motionness from dominating the training process. For clarity, we further plot the loss surface of \( \mathcal{L}_g \) in Fig. 4. Specifically, we randomly select one action category \( c \) for video \( V_n \) which has the ground truth label 1. Given the network and the trained checkpoint, we manually shift the predicted probability \( p_{n,c} \) and the mean motionness score \( \mu_{n,c} \) to compute the \( \mathcal{L}_g \) value according to Eq. (10). As shown in Fig. 4, our motion-guided loss is closely related to both the predicted probability and the video-level motionness score. To be specific, the heaviest loss penalty is applied when both \( p_{n,c} \) and \( \mu_{n,c} \) are high \( (c_1 \text{ in Fig. 4}) \). For samples with high \( p_{n,c} \) yet low \( \mu_{n,c} \) \((c_2 \text{ in Fig. 4})\), they deserve a moderate loss penalty because of their low certainty to be action-of-interest. Samples with both high \( p_{n,c} \) and \( \mu_{n,c} \) \((c_3 \text{ in Fig. 4})\) are given the slightest punishment.

### F. Inference

During testing, given an input video, we first generate its actionness scores, aggregate top-\( k \) scores, and threshold it with \( \theta_a \) to get video-level classification results. Then for the predicted action classes, we threshold the corresponding actionness scores with threshold \( \theta_a \) to select candidate snippets. Finally, we group continuous snippets into proposals and use NMS (Non-Maximum Suppression) to remove duplicates.

### IV. EXPERIMENTS

In this section, we conducted extensive experiments to demonstrate the effectiveness of our DMP-Net. Comprehensive ablation studies on the THUMOS’14 testing set were also performed to provide more insights into each component.
A. Datasets

We evaluated DMP-Net on three prevailing and challenging WSTAL benchmarks: THUMOS’14, ActivityNet v1.2 and ActivityNet v1.3. Although frame-level labels are provided in these datasets, we only use the video-level category labels for network training.

1) THUMOS’14 [4]: It is a widely-adopted benchmark for WSTAL, which consists of untrimmed videos from 20 sports categories. The training, validation, and test sets contain 13,320, 1,010, and 1,574 videos, respectively. It is very challenging since each video in this dataset may contain multiple action instances, with an average of 1.12 classes per video. Following the common setting in [7], [20], we used the validation set (200 videos) for training and the test set (213 videos) for testing.

2) ActivityNet [1]: It is a large-scale benchmark for WSTAL, which contains two versions: ActivityNet v1.3 and ActivityNet v1.2. ActivityNet v1.3 consists of 19,994 untrimmed videos from 200 classes. The training, validation, and test splits are divided by the ratio of 2:1:1. ActivityNet v1.2 is a subset of ActivityNet v1.3, which covers 100 action categories with 4,819 training, 2,383 validation and 2,480 test videos. Each video has an average of 1.65 action instances. Following the common practice [21], [28], we trained the models on the training set and evaluated them on the validation set.

B. Implementation Details

1) Evaluation Metrics: Following the standard evaluation protocol, we reported mean Average Precision (mAP) values under different Intersection over Union (IoU) thresholds. For THUMOS’14, the IoU thresholds are set to 
\[\{0.1, 0.2, \ldots, 0.9\}\]. For ActivityNet, the thresholds are chosen from 
\[\{0.5, 0.75, 0.95\}\], and we also report the average mAP of all the IoU thresholds between 0.5 and 0.95 with the step of 0.05. The evaluations of both two versions are calculated by using the official codes.

2) Feature Extractor: We used the I3D network [2] pre-trained on Kinetics [2] as our feature extractor. Note that the I3D feature extractor is not fine-tuned for fair comparisons. We applied the TVL1 [18] algorithm to extract optical flow in advance. Video snippets were sampled every 16 frames and the feature dimension for each extracted snippet was 1,024.

3) Training Details: The sampling number \(T\) was set as 750 for THUMOS’14 and 50 for ActivityNet, respectively. Following [7], [15], we performed stratified random perturbation on the segments sampled for data augmentation and used the uniform sampling strategy during the test. All hyper-parameters were determined by grid search: GCN layer number \(K = 2\), selection ratio \(r = 8\). We set the embedding dimension in GCN as 1024. For all datasets, positional threshold \(\theta_{pos}\) and similarity threshold \(\gamma\) were set to 0.1 and 0.6, respectively. We used the Adam optimizer with a learning rate of 1e-4. We trained for a total of 6k epochs with a batch size of 32 for THUMOS’14 dataset and 8k epochs with a batch size of 128 for ActivityNet dataset. Experiments were conducted on a single V100 GPU.

3) Analysis of Deep Motion Prior

In our paper, we emphasize the importance of the optical flow modality and build a deep motion prior to guide the training of the whole network. Here we design some pilot experiments to verify the necessity of our motivation.

1) Optical Flow v.s. RGB Modalities: To intuitively compare the two modalities, we conducted comparative experiments based on each single modality. Specifically, we selected two representative WSTAL methods: BaS-Net [7] and TSCN [36], which belong to early-fusion and late-fusion types, respectively. UM-Net takes the concatenation of the RGB and the optical flow modalities as input. TSCN is trained based on the two features separately and then the outputs of each branch are summed up. For both methods, we replaced the network input with only a single modality feature and retrained the network. These two model variants are called flow-based and RGB-based approaches while the original model is called the two-stream-based approach. All results are reported in TABLE IV.

   a) Quantitative results: The results in TABLE IV show that the performance of the flow-based approaches is slightly lower than the two-stream-based approaches, but far exceeds that of the RGB-based ones. For example, on mAP@IoU 0.5 of BaS-Net, the flow-based approach decreases by

   b) Qualitative results: The qualitative results on the videos show that the temporal action proposals generated by the flow-based model are more consistent with the ground truth than the RGB-based one.
only 0.6% over the official two-stream approach, while the RGB-based one has a dramatic 8.6% performance drop. The results show that the RGB modality is less sensitive to actions [36] and further verify our hypothesis that the optical flow modality is more informative than the RGB modality.

2) Learning Motion Prior from Different Modalities: In our method, we feed the optical flow modality to the guidance branch to generate the deep motion prior. Here we conducted comparison experiments that take the RGB or two-stream modality as input to the guidance branch. Besides, for more intuitive understanding, we also compute KL divergence values between the ground truth and the guidance sequence. Results are reported in TABLE V.

| Supervision (Feature) | Method | Venue | mAP@IoU (%) |
|-----------------------|--------|-------|-------------|
|                       |        |       | 0.1  | 0.2  | 0.3  | 0.4  | 0.5  | 0.6  | 0.7  | 0.8  | 0.9  |
| Full                  | S-CNN [22] | CVPR 2016 | 47.7 | 43.5 | 36.3 | 28.7 | 19.0 | 10.3 | 5.3  | -    | -    |
|                       | SSN [39]  | ICCV 2017 | 66.0 | 59.4 | 51.9 | 41.0 | 29.8 | -    | -    | -    | -    |
|                       | P-GCN [35] | ICCV 2019 | 69.5 | 67.8 | 63.6 | 57.8 | 49.1 | -    | -    | -    | -    |
|                       | S-1AD [30] | CVPR 2020 | -   | -    | -    | -    | -    | -    | 66.4 | 60.4 | 51.6 | 37.6 | 22.9 | -    | -    |
| Weak                  | Hide-and-Seek [24] | ICCV 2017 | 36.4 | 27.8 | 19.5 | 12.7 | 6.8  | -    | -    | -    | -    |
|                       | UntrimmedNet [28] | CVPR 2017 | 44.4 | 37.7 | 28.2 | 21.1 | 13.7 | -    | -    | -    | -    |
|                       | AutoLoc [21] | ECCV 2018 | -   | -    | -    | 35.8 | 29.0 | 21.2 | 13.4 | 5.8  | -    | -    | -    | -    |
|                       | CleanNet [11] | ICCV 2019 | -   | -    | -    | 37.0 | 30.9 | 23.9 | 13.9 | 7.1  | -    | -    | -    | -    |
|                       | Bas-Net [7]  | AAI 2020  | -   | -    | -    | 42.8 | 34.7 | 25.1 | 17.1 | 9.3  | -    | -    | -    | -    |
|                       | STPN [15]  | CVPR 2018 | 52.0 | 44.7 | 35.5 | 25.8 | 16.9 | 9.9  | 4.3  | 1.2  | 0.1  |
|                       | W-TALC [17] | ECCV 2018 | 55.2 | 49.6 | 40.1 | 31.1 | 22.8 | -    | 7.6  | -    | -    | -    | -    | -    |
|                       | Liu et al. [10] | CVPR 2019 | 57.4 | 50.8 | 41.2 | 32.1 | 23.1 | 15.0 | 7.0  | -    | -    | -    | -    | -    |
| Weak                  | Nguyen et al. [16] | ICCV 2019 | 60.4 | 56.0 | 46.6 | 37.5 | 26.8 | 17.6 | 9.0  | 3.3  | 0.4  |
|                       | BaS-Net [7]  | AAI 2020  | 58.2 | 52.3 | 44.6 | 36.0 | 27.0 | 18.6 | 10.4 | 3.9  | 0.5  |
|                       | DGM [20]    | CVPR 2020 | 60.0 | 54.2 | 46.8 | 38.2 | 28.8 | 19.8 | 11.4 | 3.6  | 0.4  |
|                       | UM-Net [8]  | AAI 2021  | 67.5 | 61.2 | 52.3 | 43.4 | 33.7 | 22.9 | 12.1 | -    | -    | -    | -    | -    |
|                       | SF-Net [12] | ECCV 2020 | -   | -    | -    | 52.8 | 42.2 | 30.5 | 20.6 | 12.0 | -    | -    | -    | -    |
|                       | CoLA [37]   | CVPR 2021 | 66.2 | 59.5 | 51.5 | 41.9 | 32.2 | 22.0 | 13.1 | -    | -    | -    | -    | -    |
|                       | DMP-Net (Ours) | -   | 68.1 | 61.7 | 52.9 | 44.0 | 34.2 | 23.5 | 13.1 | 4.7  | 0.6  | -    | -    | -    | -    |

Authorized licensed use limited to: University Town Library of Shenzhen. Downloaded on September 16,2022 at 03:26:19 UTC from IEEE Xplore. Restrictions apply.
TABLE V
ABLATION STUDIES ON THUMOS’14 OF DIFFERENT LOSS FUNCTIONs FOR DMP-NET. $\mathcal{L}_a$ DENOTES THE TRADITIONAL XE LOSS. $\mathcal{L}_{g,\text{Flow}}, \mathcal{L}_{g,\text{RGB}}$ AND $\mathcal{L}_{g,\text{RGB}}$ REPRESENT THE MOTION-GUIDED (OURS), RGB-GUIDED, AND TWO-STREAM GUIDED LOSSES, RESPECTIVELY. $\mathcal{L}_{g,\text{Flow}}$ DENOTES THE MOTION-GUIDED LOSS VARIANT WITHOUT THE REGULARIZATION TERM. KL DENOTES THE KL DIVERGENCE BETWEEN THE GROUND TRUTH AND THE GUIDANCE SEQUENCE.

| Loss       | mAP@IoU (%) | KL   |
|------------|-------------|------|
|            | 0.4 | 0.5 | 0.6 | 0.7 |
| $\mathcal{L}_{g,\text{Flow}}$ | 44.0 | 34.2 | 23.5 | 13.1 | 0.004 |
| $\mathcal{L}_{g,\text{RGB}}$ | 31.9 | 22.2 | 13.5 | 4.6 | 0.087 |
| $\mathcal{L}_{g,\text{RGB}}$ | 40.5 | 28.3 | 20.8 | 11.0 | 0.023 |
| $\mathcal{L}_{g,\text{Flow}}^\prime$ | 33.1 | 23.8 | 15.3 | 7.9 | - |
| $\mathcal{L}_{g,\text{Flow}}^\prime$ | 40.7 | 30.4 | 19.7 | 11.4 | 0.018 |

Fig. 5. Loss curves of XE loss (orange) and motion-guided loss (blue) for DMP-Net on THUMOS’14.

motionness sequence generated by $\mathcal{L}_{g,\text{Flow}}$ has the most correlated distributions with ground truth according to the KL divergence values. Thus, we conclude that the RGB modality is not suitable for our motionness modeling and can not guide the network training.

E. Ablation Studies on Motion-Guided Loss

Leveraging the deep motion prior, we propose a motion-guided loss to modulate the network training. Here more ablation studies are conducted to give in-depth analysis from both quantitative and qualitative perspectives.

1) Motion-Guided Loss v.s. XE Loss: We conducted comparison experiments supervised by the traditional cross-entropy loss $\mathcal{L}_a$ (cf. Sec. III-E). All other experimental settings remained unchanged except for the training loss.

a) Quantitative results: Results in TABLE V demonstrate that replacing $\mathcal{L}_{g,\text{Flow}}$ with $\mathcal{L}_a$ results in significant performance drops (e.g., 10.4% at mAP@0.5), which proves that motionness modeling is essential and the motion-guided loss can efficiently guide network training with this deep motion prior.

b) Qualitative results: We presented the loss curves of the cross-entropy loss $\mathcal{L}_a$ and motion-guided loss $\mathcal{L}_{g,\text{Flow}}$ in Fig. 5, which shows that our motion-guided loss leads to a more gentle descent process yet converges to a lower loss value.

Besides, we randomly selected one video from the THUMOS’14 testing set and visualize TCAS results of well-trained models using $\mathcal{L}_a$ and $\mathcal{L}_g$, respectively. In the SoccerPenalty video of Fig 7, the $\mathcal{L}_a$ equipped model fails to locate the 1st scene (i.e., $s_1$) maybe because the football goal does not appear in the camera shot. Besides, the mass celebration after scoring (i.e., $s_3$) is also mistaken as the action. Our DMP-Net effectively filters out these error-prone clips.

2) Ablations of Motion-Guided Loss Components: We apply a regularization term in the motion-guided loss computation, which prevents motionness from dominating the network training. The ablation studies for this term are presented in TABLE. V.

a) Quantitative results: From TABLE. V, we find that the loss variant without the regularization term has the inferior performance to the full version. For example, on mAP@IoU 0.5, the performance drops by 3.8% (19.7% v.s. 23.5%), which demonstrates that our regularization term is beneficial to the network training.

F. Analysis of Motion Graph

To obtain the context-dependent motion prior, we construct a motion graph with positional edges and semantic edges.

3More qualitative results and video demos are available in the supplementary materials.
Here we present ablation studies of the graph components and compare our sparse graph to the fully-connected one.

1) The Necessity of Modeling Relationship Between Nodes: As illustrated in Sec. III-D, we introduced graph convolution layers to help enable information dissemination among snippets. To demonstrate its efficacy, we implemented the motionness modeling with a 2-layer MultiLayer-Perceptron (MLP) for comparison. Specifically, we discarded the adjacent matrix in Eq. (4), namely using $X^k = X^{k-1}W^k$ for each layer updating, where $W^k$ are learnable parameters.

a) Quantitative results: As shown in TABLE VI, GCN leads to better performance at all IoUs, which justifies its superiority in the message passing among snippets.

b) Qualitative results: We also chose one video and visualized the TCAS and motionness distributions for two variants using GCN and MLP, respectively3. The results in Fig. 6 demonstrate that GCN helps capture action-of-interest and get rid of the distracting actions, e.g., player dribbling is mistakenly highlighted by MLP, and the corresponding TCAS is dominated by such misleading background actions.

2) Ablations Studies on Positional and Semantic Edges: Positional edges and semantic edges are designed to capture surrounding and remote yet semantically related nodes, respectively. We conducted ablation studies on the graph components, and all results are reported in Table VI.

a) Quantitative results: As shown in Table VI, removing any type of edge leads to the remarkable performance drop, especially for the semantic edges. For example, mAP@IoU 0.6 drops by 1.9% in the absence of $E_{sem}$. Thus both kinds of edges play a very important role in relationship modeling.

b) Qualitative results: An intuitive way to build the motion graph is to connect all nodes in a dense manner, i.e., edges are built between all possible node pairs. Specifically, following Eq. (5), we computed the adjacent matrix as $G_{ij} = \frac{(W_1m_j)(W_2m_j)}{\sum_i(W_1m_i)(W_2m_i)}$, where $W_1$ and $W_2$ are learnable parameters and $T$ is the number of temporal nodes. Obviously, this fully-connected approach is more computation intensive and requires more storage resources.

a) Quantitative results: As shown in TABLE VI, the performance of the fully-connected graph is behind our sparse graph (i.e., w/ all edges). For example, the performance drops by 1.4% at mAP@IoU 0.5 (34.2% v.s. 32.8%).

b) Qualitative results: To reveal the rationale behind this, we selected one video from the THUMOS'14 testing set and visualized the adjacent matrix of both the dense connected and our sparse graph in Fig. 8. We can easily observe that the high weight of the adjacency matrix of the fully-connected graph is mainly concentrated in the diagonal area, i.e., each node is more relevant to its surrounding nodes. This may be due to the slowness prior [38] of video data, where short-term features change slowly in a local window. Therefore, the adjacent weight is focused on the positional adjacent areas while neglecting the remote yet semantic correlated snippets. In contrast, our sparse graph alleviates this problem by encouraging the semantic correlations of snippets even when they are far away. For example, in Fig. 8, the Shotput action conducted by two different actors is scattered in a long temporal range. Our motion graph can capture these semantic connections while the dense connection way ignores them.

G. Analysis of Model Scalability

Our proposed motion-guided loss can be easily integrated into other WSTAL frameworks as an alternative to the XE loss. Thus, we conducted more experiments to demonstrate the model scalability.

1) More Experiments on Other Backbones: To prove the general effectiveness of our motion-guided loss, in addition to

| Mode         | mAP@IoU (%) |
|--------------|-------------|
| MLP          | 39.1 30.2 19.2 10.4 |
| Dense        | 41.0 32.8 21.5 11.3 |
| w/ all edges | 44.0 34.2 23.5 13.1 |
| Sparse       | 42.7 33.6 22.7 12.4 |
| w/o $E_{pos}$| 41.5 33.3 21.6 11.5 |

| Model      | mAP@IoU (%) |
|------------|-------------|
| BaS-Net    | 36.0 27.0 18.6 10.4 |
| BaS-Net*   | 39.8 31.2 20.4 11.2 |
| UM-Net     | 43.4 37.7 22.5 12.1 |
| UM-Net*    | 44.2 34.5 23.2 12.8 |

| $\theta_{pos}$ | mAP@0.5 | 0.05 | 0.1 0.2 0.3 |
|-----------------|---------|------|--------------|
| 0.01            | 33.5    | 33.9 | 34.2 33.0 32.8 |

| $\gamma$       | mAP@0.5 | 34.3 | 34.2 32.1 31.4 |
|-----------------|---------|------|--------------|
| 0.4             | 34.3    | 34.2 | 32.1 31.4    |
the baseline backbone applied above, we selected two typical WSTAL models (BaS-Net [7] and UM-Net [8]) and replaced the original XE loss with our proposed motion guided loss. Specifically, for both methods, we additionally introduced our guidance branch and used it to guide the network training in the format of motion-guided loss. All the hyper-parameter settings remained the same as in our baseline. The results are reported in TABLE VII.

a) Quantitative results: As shown in TABLE VII, our motion-guided loss leads to consistent improvement for both methods. For example, in BaS-Net, mAP@IoU 0.5 is improved by 4.2% after applying our motion-guided loss. These results demonstrate that our motion-guided loss is general and compatible with different backbones.

2) Analysis of the Hyper-Parameter Sensitivity: We conducted experiments on two important hyper-parameters including the positional threshold $\theta_{pos}$ and similarity threshold $\gamma$. 

a) Quantitative results: As shown in TABLE VIII, mAP@0.5 hits the peak performance when setting $\theta_{pos} = 0.1$. We have noticed that too large or too small values of $\theta_{pos}$ both lead to performance degradation. This may be because a small $\theta_{pos}$ value will affect the modeling of local correlations while too large $\theta_{pos}$ value will make the network similar to the fully-connected graph, which has been demonstrated to be a sub-optimal design. For the value of $\gamma$, we see that when $\gamma$ is greater than 0.6, performance degrades rapidly. On the other hand, the cost of the number of edges decreases as the value of $\gamma$ increases. To trade-off between both the accuracy and the computation overhead comprehensively, we set $\gamma = 0.6$.

3) Analysis of Model Complexity: We conduct the model complexity analysis for both baseline and our DMP-Net. We set up three evaluation metrics, i.e., #param, GFLOPs, and run-time. Run-time is defined as the average inference time to localize one untrimmed video.

As shown in TABLE IX, DMP-Net introduces extra computational costs during the training process. However, considering the significant performance gains, these costs are acceptable. Besides, DMP-Net shares the same inference process as the baseline model and therefore the introduced guidance branch does not affect the reasoning speed.

V. CONCLUSION

In this paper, we start from summarizing the two overlooked issues in existing Weakly-Supervised Temporal Action Localization (WSTAL) methods, i.e., inadequate use of optical flow modality and the incompatibility of XE loss. Then, we analyze and argue that effective motion modeling is essential in WSTAL. Accordingly, we apply GCNs on optical flow to obtain a context-dependent motion prior, termed as motionness. Besides, we use it to modulate the video-level classification, yielding a novel Motion-guided Loss. Experiments conducted on three benchmarks including THUMOS’14, ActivityNet v1.2, and ActivityNet v1.3 datasets have validated the state-of-the-art performance of our proposed DMP-Net.

REFERENCES

[1] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 961–970.
[2] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6298–6308.
[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, “A survey on visual content-based video indexing and retrieving,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 41, no. 6, pp. 797–819, Nov. 2011.
[4] H. Idrees et al., “The THUMOS challenge on action recognition for videos in the wild,” Comput. Vis. Image Understand., vol. 155, pp. 1–23, Feb. 2017.
[5] S. Jiang, S. Liang, C. Chen, Y. Zhu, and X. Li, “Class agnostic image common object detection,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2836–2846, Jun. 2019.
[6] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.
[7] P. Lee, Y. Uh, and H. Byun, “Background suppression network for weakly-supervised temporal action localization,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 11320–11327.
[8] P. Lee, J. Wang, Y. Lu, and H. Byun, “Weakly-supervised temporal action localization by uncertainty modeling,” in Proc. AAAI Conf. Artif. Intell., vol. 2, 2021, pp. 1–9.
[9] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1346–1353.
[10] D. Liu, T. Jiang, and Y. Wang, “Completeness modeling and context separation for weakly supervised temporal action localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1298–1307.
[11] Z. Liu et al., “Weakly supervised temporal action localization through contrast based evaluation networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3899–3908.
[12] F. Ma et al., “SF-Net: Single-frame supervision for temporal action localization,” in Proc. Eur. Conf. Comput. Vis. Springer, 2020, pp. 420–437.
[13] K. Min and J. J. Corso, “Adversarial background-aware loss for weakly-supervised temporal activity localization,” in Proc. Eur. Conf. Comput. Vis. Springer, 2020, pp. 283–299.
[14] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, “3C-Net: Category count and center loss for weakly-supervised action localization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 8679–8687.
[15] P. Nguyen, B. Han, T. Liu, and G. Prasad, “Weakly supervised action localization by sparse temporal pooling network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6752–6761.
[16] P. Nguyen, D. Ramanan, and C. Fowlkes, “Weakly-supervised action localization with background modeling,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5502–5511.
[17] S. Paul, S. Roy, and A. K. Roy-Chowdhury, “W-TALC: Weakly-supervised temporal activity localization and classification,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 563–579.
[18] J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo, “TV-L1 optical flow estimation,” Image Process. OnLine, vol. 2013, pp. 137–150, Jul. 2013.
[19] M. Rashid, H. Kjellstrom, and Y. J. Lee, “Action graphs: Weakly-supervised action localization with graph convolution networks,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2020, pp. 615–624.
[20] B. Shi, Q. Dai, Y. Mu, and J. Wang, “Weakly-supervised action localization by generative attention modeling,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1009–1019.
[21] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, “AutoLoc: Weakly-supervised temporal action localization in untrimmed videos,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 154–171.
[22] Z. Shou, D. Wang, and S.-F. Chang, “Temporal action localization in untrimmed videos via multi-stage CNNs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1049–1058.
[23] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 568–576.
[24] K. K. Singh and Y. J. Lee, “Hide- and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3544–3553.

[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 4489–4497.

[26] S. Vishwakarma and A. Agrawal, “A survey on activity recognition and behavior understanding in video surveillance,” Vs. Comput., vol. 29, no. 10, pp. 983–1009, Oct. 2013.

[27] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Han, “Graph attention convolution for point cloud semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 10296–10305.

[28] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, “UntrimmedNets for weakly supervised action recognition and detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4325–4334.

[29] H. Xu, C. Jiang, X. Liang, and Z. Li, “Spatial-aware graph relation network for large-scale object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9298–9307.

[30] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, “G-TAD: Sub-graph localization for temporal action detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10156–10165.

[31] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1–10.

[32] T. Yu, Z. Ren, Y. Li, E. Yan, N. Xu, and J. Yuan, “Temporal structure mining for weakly supervised action detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5522–5531.

[33] Y. Yuan, Y. Lyu, X. Shen, I. W. Tsang, and D.-Y. Yeung, “Marginalized average attentional network for weakly-supervised learning,” 2019, arXiv:1905.08586.

[34] R. Zeng, C. Gan, P. Chen, W. Huang, Q. Wu, and M. Tan, “Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization,” IEEE Trans. Image Process., vol. 28, no. 12, pp. 5797–5808, Dec. 2019.

[35] R. Zeng et al., “Graph convolutional networks for temporal action localization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7094–7103.

[36] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, and G. Hua, “Two-stream consensus network for weakly-supervised temporal action localization,” in Proc. Eur. Conf. Comput. Vis. Springer, 2020, pp. 37–54.

[37] C. Zhang, M. Cao, D. Yang, J. Chen, and Y. Zou, “CoLA: Weakly-supervised temporal action localization with snippet contrastive learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16010–16019.

[38] Z. Zhang and D. Tao, “Slow feature analysis for human action recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 3, pp. 436–450, Mar. 2012.

[39] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, “Temporal action detection with structured segment networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2914–2923.

[40] J.-X. Zhong, N. Li, W. Kong, T. Zhang, T. H. Li, and G. Li, “Step-by-step erasure, one-by-one collection: A weakly supervised temporal action detector,” in Proc. 26th ACM Int. Conf. Multimedia, Oct. 2018, pp. 35–44.

[41] L. Zhu, H. Fan, Y. Luo, M. Xu, and Y. Yang, “Few-shot common-object reasoning using common-centric localization network,” IEEE Trans. Image Process., vol. 30, pp. 4253–4262, 2021.

Meng Cao received the B.Eng. degree from the Huazhong University of Science and Technology (HUST) in 2018. He is currently pursuing the Ph.D. degree with the School of Electronic and Computer Engineering (ECE), Peking University, Shenzhen, China. His current research interests include computer vision and computer graphics.

Can Zhang received the B.Eng. degree in electronic engineering from Shanghai University, Shanghai, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Electronic and Computer Engineering (ECE), Peking University, Shenzhen, China. His current research interests include computer vision, multimedia computing, and video understanding.

Long Chen received the B.Eng. degree in electrical information engineering from the Dalian University of Technology, Dalian, China, in 2015, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2020. He was a Senior Researcher at Tencent AI Lab, Shenzhen, China. He is currently a Postdoctoral Research Scientist at Columbia University, New York, USA, working with Prof. Shih-Fu Chang. His research interests are computer vision, machine learning, and multimedia.

Mike Zheng Shou (Member, IEEE) received the Ph.D. degree from the Columbia University in the City of New York. He was a Research Scientist at Facebook AI in Bay Area. He is currently an Assistant Professor (tenure-track) at the National University of Singapore. He is a fellow of the National Research Foundation (NRF), Singapore, Class of 2021. He was awarded the Wei Family Private Foundation Fellowship. He has received the Best Student Paper Nomination at CVPR’17. His team has won the First Place in the International Challenge on Activity Recognition (ActivityNet) 2017.

Yuexian Zou received the M.Sc. degree from the University of Electronic Science and Technology of China in 1991 and the Ph.D. degree from The University of Hong Kong in 2000. She is currently a Full Professor with Peking University and the Director of the Advanced Data and Signal Processing Laboratory, Peking University Shenzhen Graduate School. Her research interests mainly in machine learning for signal processing and deep learning and its applications. She was a recipient of the award Leading Figure for Science and Technology by Shenzhen Municipal Government in 2009.