BMN: Boundary-Matching Network for Temporal Action Proposal Generation

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, Shilei Wen
Department of Computer Vision Technology (VIS), Baidu Inc.
{lintianwei01,liuxiao12,lixin41,dingerrui,wenshilei}@baidu.com

Abstract

Temporal action proposal generation is an challenging and promising task which aims to locate temporal regions in real-world videos where action or event may occur. Current bottom-up proposal generation methods can generate proposals with precise boundary, but cannot efficiently generate adequately reliable confidence scores for retrieving proposals. To address these difficulties, we introduce the ***Boundary-Matching (BM) mechanism*** *to evaluate confidence scores of densely distributed proposals, which denote a proposal as a matching pair of starting and ending boundaries and combine all densely distributed BM pairs into the BM confidence map. Based on BM mechanism, we propose an effective, efficient and end-to-end proposal generation method, named* ***Boundary-Matching Network (BMN)****, which generates proposals with precise temporal boundaries as well as reliable confidence scores simultaneously. The two-branches of BMN are jointly trained in an unified framework. We conduct experiments on two challenging datasets: THUMOS-14 and ActivityNet-1.3, where BMN shows significant performance improvement with remarkable efficiency and generalizability. Further, combining with existing action classifier, BMN can achieve state-of-the-art temporal action detection performance.*

1. Introduction

With the number of videos in Internet growing rapidly, video content analysis methods have attracted widespread attention from both academia and industry. Temporal action detection is an important task in video content analysis area, which aims to locate action instances in untrimmed long videos with both action categories and temporal boundaries. Akin to object detection, temporal action detection method can be divided into two stages: temporal action proposal generation and action classification. Although convincing classification accuracy can be achieved by action recognition methods, the detection performance is still low in mainstream benchmarks [15, 5]. Therefore, many recent methods work on improving the quality of temporal action proposals. Besides being used in temporal action detection task, temporal proposal generation methods also have wide applications in many areas such as video recommendation, video highlight detection and smart surveillance.

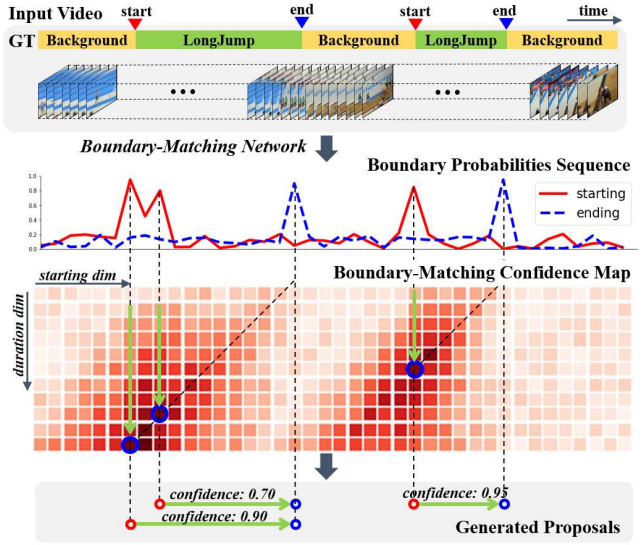

Figure 1. Overview of our method. Given an untrimmed video, BMN can simultaneously generate (1) boundary probabilities sequence to construct proposals and (2) Boundary-Matching confidence map to densely evaluate confidence of all proposals.

To achieve high proposal quality, a proposal generation method should (1) generate temporal proposals with flexible duration and precise boundaries to cover ground-truth action instances precisely and exhaustively; (2) generate reliable confidence scores so that proposals can be retrieved properly. Most existing proposal generation methods [3, 4, 8, 24] adopted a **"top-down"** fashion to generate proposals with multi-scale temporal sliding windows in regular interval, and then evaluate confidence scores of proposals respectively or simultaneously. The main drawback of these methods is that generated proposals are usually not temporally precise or not flexible enough to cover ground-truth action instances of varies duration. Recently, Boundary-Sensitive Network (BSN) [18] adopted a **"bottom-up"** fashion to generate proposals in two stages:

(1) locate temporal boundaries and combine boundaries as proposals and (2) evaluate confidence score of each proposal using constructed proposal feature. By exploiting local clues, BSN can generate proposals with more precise boundaries and more flexible duration than existing top-down methods. However, BSN has three main drawbacks: (1) proposal feature construction and confidence evaluation procedures are conducted to each proposal respectively, leading to inefficiency; (2) the proposal feature constructed in BSN is too simple to capture enough temporal context; (3) BSN is multiple-stage but not an unified framework.

Can we evaluate confidence for all proposals simultaneously with rich context? Top-down methods [19, 2] can achieve this easily with anchor mechanism, where proposals are pre-defined as non-continuous distributed anchors. However, since the boundary and duration of proposals are much more flexible, anchor mechanism is not suitable for bottom-up methods such as BSN. To address these difficulties, we propose the **Boundary-Matching (BM) mechanism** for confidence evaluation of densely distributed proposals. In BM mechanism, a proposal is denoted as a matching pair of its starting and ending boundaries, and then all BM pairs are combined as a two dimensional BM confidence map to represent densely distributed proposals with continuous starting boundaries and temporal duration. Thus, we can generate confidence scores for all proposals simultaneously via the BM confidence map. A BM layer is proposed to generate BM feature map from temporal feature sequence, and the BM confidence map can be obtained from the BM feature map using a series of conv-layers. BM feature map contains rich feature and temporal context for each proposal, and gives the potential for exploiting context of adjacent proposals. Codes are avaiable at PaddleVideo.

In summary, our work has three main contributions:

1. We introduce the *Boundary-Matching mechanism* for evaluating confidence scores of densely distributed proposals, which can be easily embedded in network.
2. We propose an efficient, effective and end-to-end temporal action proposal generation method *Boundary-Matching Network* (BMN). Temporal boundary probability sequence and BM confidence map are generated simultaneously in two branches of BMN, which are trained jointly as an unified framework.
3. Extensive experiments show that BMN can achieve significantly better proposal generation performance than other state-of-the-art methods, with remarkable efficiency, great generalizability and great performance on temporal action detection task.

2. Related Work

Action Recognition. Action recognition is a fundamental and important task of video understanding area. Handcrafted features such as HOG, HOF and MBH are widely used in earlier works, such as improved Dense Trajectory (iDT) [29, 30]. Recently, deep learning models have achieved significantly performance promotion in action recognition task. The mainstream networks fall into two categories: two-stream networks [9, 25, 32] exploit appearance and motion clues from RGB image and stacked optical flow separately; 3D networks [27, 22] exploit appearance and motion clues directly from raw video volume. In our work, by convention, we adopt action recognition models to extract visual feature sequence of untrimmed video.

Correlation Matching. Correlation matching algorithms are widely used in many computer vision tasks, such as image registration, action recognition and stereo matching. Specifically, stereo matching aims to find corresponding pixels from stereo images. For each pixel in left image of a rectified image pair, the stereo matching method need to find corresponding pixel in right image along horizontal direction, or we can say finding right pixel with minimum cost. Thus, the cost minimization of all left pixels can be denoted as a cost volume, which denotes each left-right pixel pair as a point in volume. Based on cost volume, many recent works [26, 21, 17] achieve end-to-end network via generating cost volume directly from combining two feature maps, using correlation layer [21] or feature concatenation [6]. Inspired by cost volume, our proposed BM confidence map contains pairs of temporal starting and ending boundaries as proposals, thus can directly generate confidence scores for all proposals using convolutional layers. We propose BM layer to efficiently generate BM feature map via sampling feature among starting and ending boundaries of each proposal simultaneously.

Temporal Action Proposal Generation. As aforementioned, the goal of temporal action detection task is to detect action instances in untrimmed videos with temporal boundaries and action categories, which can be divided into temporal proposal generation and action classification stages. These two stages are taken apart in most detection methods [24, 36, 35], and are taken together as single model in some methods [19, 2, 14]. For proposal generation task, most previous works [3, 4, 8, 12, 24] adopt *top-down* fashion to generate proposals with pre-defined duration and interval, where the main drawback is the lack of boundary precision and duration flexibility. There are also some methods [36, 18] adopt *bottom-up* fashion. TAG [36] generates proposals using temporal watershed algorithm, but lack confidence scores for retrieving. Recently, BSN [18] generates proposals via locally locating temporal boundaries and globally evaluating confidence scores, and achieves significant performance promotion over previous proposal generation methods. In this work, we propose the Boundary-Matching mechanism for proposal confidence evaluation, which can largely simplify the pipeline of BSN and bring significant promotion in both efficiency and effectiveness.

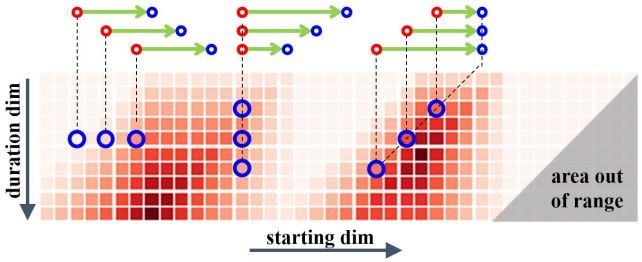

Figure 2. Illustration of BM confidence map. Proposals in the same row have the same temporal duration, and proposals in the same column have the same starting time. The ending boundaries of proposals at right-bottom corner exceed the range of video, thus these proposals are not considered during training and inference.

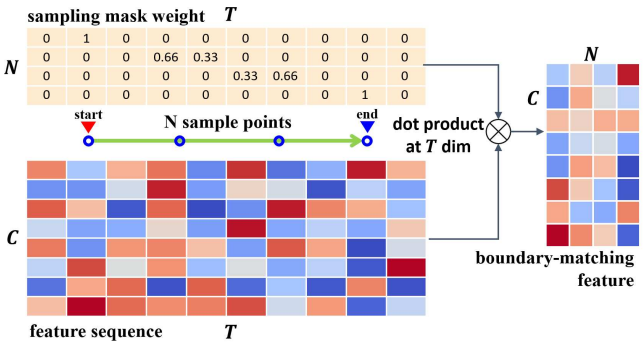

Figure 3. Illustration of BM layer. For each proposal, we conduct dot product at T dimension between sampling weight and temporal feature sequence, to generate BM feature of shape $C \times N$.

.

3. Our Approach

3.1. Problem Formulation

We can denote an untrimmed video X as frame sequence $X = \{x_n\}_{n=1}^{l_v}$ with l_v frames, where x_n is the n-th RGB frame of video X. The temporal annotation set of X is composed by a set of temporal action instances as $\Psi_g = \{\varphi_n = (t_{s,n}, t_{e,n})\}_{n=1}^{N_g}$, where N_g is the amount of ground-truth action instances, $t_{s,n}$ is the starting time of action instance φ_n and $t_{e,n}$ is the ending time. Unlike temporal action detection task, categories of action instances are not taken into account in proposal generation task. During inference, proposal generation method should generate proposals Ψ_p which cover Ψ_g precisely and exhaustively.

3.2. Feature Encoding.

Following recent proposal generation methods [3, 8, 12, 18], we construct BMN model upon visual feature sequence extracted from raw video. In this work, we adopt two-stream network [25] for feature encoding since it achieves great action recognition precision and is widely used in many video analysis methods [11, 19, 36]. Concatenating the output scores of top fc-layer in two-stream network, we can get encoded visual feature $f_{t_n} \in R^C$ around frame x_{t_n}, where C is the dimension of feature. Therefore, given an untrimmed video X of length l_v, we can extract a visual feature sequence $F = \{f_{t_n}\}_{n=1}^{l_f} \in R^{C \times l_f}$ with length l_f. To reduce the computation cost, we extract feature in a regular frame interval σ, thus $l_f = l_v/\sigma$.

3.3. Boundary-Matching Mechanism

In this section, we introduce the Boundary-Matching (BM) mechanism to generate confidence scores for densely distributed proposals. First we denote a temporal proposal φ as a matching pair of its starting boundary t_s and ending boundary t_e. Then, as shown in Fig 2, the goal of BM mechanism is to generate the two dimensional BM confidence map M_C, which is constructed by BM pairs with different starting boundary and temporal duration. In BM confidence map, the value of point $M_C(i,j)$ is denoted as the confidence score of proposal $\varphi_{i,j}$ with starting boundary $t_s = t_j$, duration $d = t_i$ and ending boundary $t_e = t_j + t_i$. Thus, we can generate confidence scores for densely distributed proposals via generating BM confidence map.

Boundary-Matching Layer. How can we generate two dimensional BM confidence map from temporal feature sequence? In BM mechanism, we introduce the BM layer to generate BM feature map $M_F \in R^{C \times N \times D \times T}$ from temporal feature sequence $S_F \in R^{C \times T}$, and then use M_F to generate BM confidence map $M_C \in R^{D \times T}$ with a series of convolutional layers, where D are pre-defined maximum proposal duration. The goal of BM layer is to uniformly sample N points in S_F between starting boundary t_s and ending boundary t_e of each proposal $\varphi_{i,j}$, and get proposal feature $m_{i,j}^f \in R^{C \times N}$ with rich context. And we can generate BM feature map M_F via conducting this sampling procedure for all proposals simultaneously.

There are two difficulties to achieve this feature sampling procedure: (1) how to sample feature in non-integer point and (2) how to sample feature for all proposals simultaneously. As shown in Fig 3, we achieve this via dot product between temporal feature sequence $S_F \in R^{C \times T}$ and sampling mask weight $W \in R^{N \times T \times D \times T}$ in temporal dimension. In detail, **first**, for each proposal $\varphi_{i,j}$, we construct weight term $w_{i,j} \in R^{N \times T}$ via uniformly sampling N points between expanded temporal region $[t_s - 0.25d, t_e + 0.25d]$. For a non-integer sampling point t_n, we define its corresponding sampling mask $w_{i,j,n} \in R^T$ as

$$w_{i,j,n}[t] = \begin{cases} 1 - dec(t_n) & \text{if } t = floor(t_n) \\ dec(t_n) & \text{if } t = floor(t_n) + 1, \\ 0 & \text{if } t = others \end{cases} \quad (1)$$

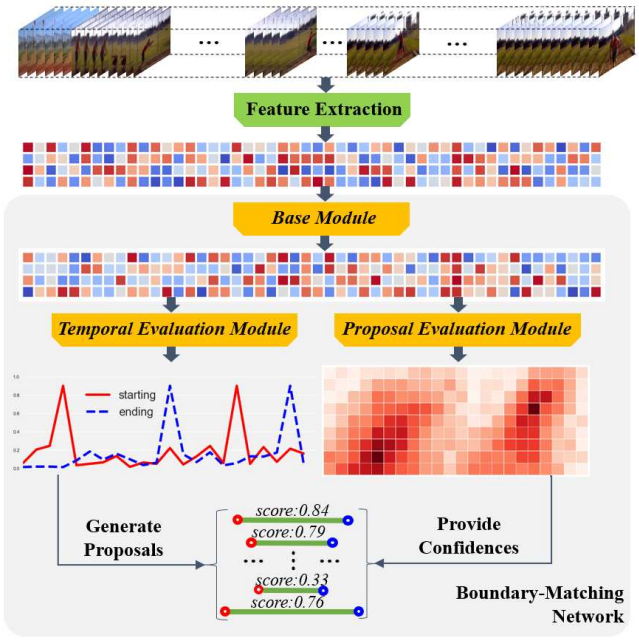

Figure 4. The framework of Boundary-Matching Network. After feature extraction, we use BMN to simultaneously generate temporal boundary probability sequence and BM confidence map, and then construct proposals based on boundary probabilities and get corresponding confidence score from BM confidence map.

where dec and $floor$ is decimal and integer fraction functions separately. Thus, for proposal $\varphi_{i,j}$, we can get weight term $w_{i,j} \in R^{N \times T}$. **Second**, we conduct dot product in temporal dimension between S_F and $w_{i,j}$

$$m^f_{i,j}[c,n] = \sum_{t=1}^{T} S_f[c,t] \cdot w_{i,j}[n,t]. \tag{2}$$

Via expanding $w_{i,j} \in R^{N \times T}$ to $W \in R^{N \times T \times D \times T}$ for all proposals in BM confidence map, we can generate BM feature map $M_F \in R^{C \times N \times D \times T}$ using dot product. Since the sampling mask weight W is the same for different videos and can be pre-generated, the inference speed of BM layer is very fast. BM feature map contains rich feature and temporal context for each proposal, and gives the potential for exploiting context of adjacent proposals.

Boundary-Matching Label. During training, we denote the BM label map as $G_C \in R^{D \times T}$ with the same shape of BM confidence map M_C, where $g^c_{i,j} \in [0,1]$ represents the maximum IoU between proposal $\varphi_{i,j}$ and all ground-truth action instances. Generally, in BM mechanism, we use BM layer to efficiently generate BM feature map M_F from temporal feature sequence S_F, and then use a series of convolutional layers to generate BM confidence map M_C, which is trained under supervision of BM label map G_C.

Table 1. The detailed architecture of BMN, where the output feature sequence of base module is shared by temporal evaluation and proposal evaluation modules. T and D are length of input feature sequence and maximum proposal duration separately.

layer	kernel	stride	dim	act	output size
Base Module					
$conv1d_1$	3	1	256	$relu$	$256 \times T$
$conv1d_2$	3	1	128	$relu$	$128 \times T$
Temporal Evaluation Module					
$conv1d_3$	3	1	256	$relu$	$256 \times T$
$conv1d_4$	3	1	2	$sigmoid$	$2 \times T$
Proposal Evaluation Module					
BM layer	N - 32				$128 \times 32 \times D \times T$
$conv3d_1$	32,1,1	32,0,0	512	$relu$	$512 \times 1 \times D \times T$
squeeze					$512 \times D \times T$
$conv2d_1$	1,1	0,0	128	$relu$	$128 \times D \times T$
$conv2d_2$	3,3	1,1	128	$relu$	$128 \times D \times T$
$conv2d_3$	1,1	0,0	2	$sigmoid$	$2 \times D \times T$

3.4. Boundary-Matching Network

Different with the multiple-stage framework of BSN [18], BMN generates local boundary probabilities sequence and global proposal confidence map simultaneously, while the whole model is trained in an unified framework. As demonstrated in Fig 4, BMN model contains three modules: *Base Module* handles the input feature sequence, and outputs feature sequence shared by the following two modules; *Temporal Evaluation Module* evaluates starting and ending probabilities of each location in video to generate boundary probability sequences; *Proposal Evaluation Module* contains the BM layer to transfer feature sequence to BM feature map, and contains a series of 3D and 2D convolutional layers to generate BM confidence map.

Base Module. The goal of the base module is to handle the input feature sequence, expand the receptive field and serve as backbone of network, to provide a shared feature sequence for TEM and PEM. Since untrimmed videos have uncertain temporal length, we adopt a long observation window with length l_ω to truncate the untrimmed feature sequence with length l_f. We denote an observation window as $\omega = \{t_{\omega,s}, t_{\omega,e}, \Psi_\omega, F_\omega\}$, where $t_{\omega,s}$ and $t_{\omega,e}$ are the starting and ending time of ω separately, Ψ_ω and F_ω are annotations and feature sequence within the window separately. The window length $l_\omega = t_{\omega,e} - t_{\omega,s}$ is set depending on the dataset. The details of base module is shown in Table 1, including two temporal convolutional layers.

Temporal Evaluation Module (TEM). The goal of TEM is to evaluate the starting and ending probabilities for all temporal locations in untrimmed video. These boundary probability sequences are used for generating proposals during post processing. The details of TEM are shown in Table 1, where $conv1d_4$ layer with two sigmoid activated filters output starting probability sequence $P_{S,\omega} = \{p^s_{t_n}\}_{n=1}^{l_\omega}$ and

ending probability sequence $P_{E,\omega} = \left\{p^e_{t_n}\right\}_{n=1}^{l_\omega}$ separately for an observation window ω.

Proposal Evaluation Module (PEM). The goal of PEM is to generate Boundary-Matching (BM) confidence map, which contains confidence scores for densely distributed proposals. To achieve this, PEM contains BM layer and a series of 3d and 2d convolutional layers.

As introduced in Sec. 3.3, BM layer transfers temporal feature sequence S to BM feature map M_F via matrix dot product between S and sampling mask weight W in temporal dimension. In BM layer, the number of sample points N is set to 32, and the maximum proposal duration D is set depending on dataset. After generating BM feature map M_F, first we conduct $conv3d_1$ layer in sample dimension to reduce dimension length from N to 1, and increase hidden units from 128 to 512. Then, we conduct $conv2d_1$ layer with 1×1 kernel to reduce the hidden units, and $conv2d_2$ layer with 3×3 kernel to capture context of adjacent proposals. Finally, we generate two types of BM confidence map $M_{CC}, M_{CR} \in R^{D \times T}$ with *sigmoid* activation, where M_{CC} and M_{CR} are trained using binary classification and regression loss function separately.

3.5. Training of BMN

In BMN, TEM learns local boundary context and PEM pattern global proposal context. To jointly learn local pattern and global pattern, an unified multi-task framework is exploited for optimization. The training details of BMN are introduced in this section.

Training Data Construction. Given an untrimmed video X, we can extract feature sequence F with length l_f. Then, we use observation windows with length l_ω to truncate feature sequence with 50% overlap, where windows containing at least one ground-truth action instance are kept for training. Thus, a training set $\Omega = \{\omega_n\}_{n=1}^{N_\omega}$ is constructed with N_ω observation windows.

Label Assignment. For TEM, we need to generate temporal boundary label sequence $G_S, G_E \in R^T$. Following BSN[18], for a ground-truth action instance $\varphi_g = (t_s, t_e)$ with duration $d_g = t_e - t_s$ in annotation set Ψ_ω, we denote its starting and ending regions as $r_S = [t_s - d_g/10, t_s + d_g/10]$ and $r_E = [t_e - d_g/10, t_e + d_g/10]$ separately. Then, for a temporal location t_n within F_ω, we denote its local region as $r_{t_n} = [t_n - d_f/2, t_n + d_f/2]$, where $d_f = t_n - t_{n-1}$ is the temporal interval between two locations. Then we calculate overlap ratio IoR of r_{t_n} with r_S and r_E separately, and denote maximum IoR as $g^s_{t_n}$ and $g^e_{t_n}$ separately, where IoR is defined as the overlap ratio with groundtruth proportional to the duration of this region. Thus we can generate $G_{S,\omega} = \left\{g^s_{t_n}\right\}_{n=1}^{l_\omega}$ and $G_{E,\omega} = \left\{g^e_{t_n}\right\}_{n=1}^{l_\omega}$ as label of TEM.

For PEM, we need to generate BM label map $G_C \in R^{D \times T}$. For a proposal $\varphi_{i,j} = (t_s = t_j, t_e = t_j + t_i)$, we calculate its Intersection-over-Union (IoU) with all φ_g in Ψ_ω, and denote the maximum IoU as $g^c_{i,j}$. Thus we can generate $G_C = \left\{g^c_{i,j}\right\}_{i,j=1}^{D,l_\omega}$ as label of PEM.

Loss of TEM. With generated boundary probability sequence $P_{S,\omega}$, $P_{E,\omega}$ and boundary label sequence $G_{S,\omega}$, $G_{E,\omega}$, we can construct the loss function of TEM as the sum of staring and ending losses

$$L_{TEM} = L_{bl}(P_S, G_S) + L_{bl}(P_E, G_E). \tag{3}$$

Following BSN[18], we adopt weighted binary logistic regression loss function L_{bl} for both starting and ending losses, where $L_{bl}(P, G)$ is denoted as:

$$\frac{1}{l_\omega} \sum_{i=1}^{l_\omega} \left(\alpha^+ \cdot b_i \cdot log(p_i) + \alpha^- \cdot (1 - b_i) \cdot log(1 - p_i)\right), \tag{4}$$

where $b_i = sign(g_i - \theta)$ is a two-value function used to convert g_i from $[0, 1]$ to $\{0, 1\}$ based on overlap threshold $\theta = 0.5$. Denoting $l^+ = \sum b_i$ and $l^- = l_\omega - l^+$, the weighted terms are $\alpha^+ = \frac{l_w}{l^+}$ and $\alpha^- = \frac{l_w}{l^-}$.

Loss of PEM. With generated BM confidence map M_{CC}, M_{CR} and BM label map G_C, we can construct the loss function of PEM, which is the sum of binary classification loss and regression loss:

$$L_{PEM} = L_C(M_{CC}, G_C) + \lambda \cdot L_R(M_{CR}, G_C). \tag{5}$$

where we adopt L_{bl} for classification loss L_C and L2 loss for regression loss L_R, and set the weight term $\lambda = 10$. To balance the ratio between positive and negative samples in L_R, we take all points with $g^c_{i,j} > 0.6$ as positive and randomly sample $g^c_{i,j} < 0.2$ as negative, and ensure the ratio between positive and negative points nearly 1:1.

Training Objective. We train BMN in the form of a multi-task loss function, including TEM loss, PEM loss and L2 regularization term:

$$L = L_{TEM} + \lambda_1 \cdot L_{PEM} + \lambda_2 \cdot L_2(\Theta), \tag{6}$$

where weight term λ_1 and λ_2 are set to 1 and 0.0001 separately to ensure different modules are trained evenly.

3.6. Inference of BMN

During inference, we use BMN to generate boundary probability sequences G_S, G_E and BM confidence map M_{CC}, M_{CR}. To get final results, we need to (1) generate candidate proposals using boundary probabilities, (2) fuse boundary probability and confidence score to generate final confidence score, (3) and suppress redundant proposals based on final confidence scores.

Candidate Proposals Generation. Following BSN [18], we generate candidate proposals via combining temporal locations with high boundary probabilities. First, to locate

high starting probability locations, we record all temporal locations t_n with starting $p^s_{t_n}$ (1) higher than $0.5 \cdot max(p)$ or (2) being a probability peak, where $max(p^s)$ is the maximum starting probability of this video. These candidate starting locations are grouped as $B_S = \{t_{s,i}\}_{i=1}^{N_S}$. We can generate ending locations set B_E in the same way.

Then we match each starting location t_s in B_S and ending location t_e in B_E as a proposal, if its duration is smaller than a pre-defined maximum duration D. The generated proposal φ is denoted as $\varphi = (t_s, t_e, p^s_{t_s}, p^e_{t_e}, p_{cc}, p_{cr})$, where $p^s_{t_s}$, $p^e_{t_e}$ are starting and ending probabilities in t_s and t_e separately, and p_{cc}, p_{cr} are classification confidence score and regression confidence score from $[t_e - t_s, t_s]$ point of BM confidence map M_{CC} and M_{CR} separately. Thus we can get candidate proposals set $\Psi = \{\varphi_i\}_{i=1}^{N_p}$, where N_p is the number of candidate proposals.

Score Fusion. To generate more reliable confidence scores, for each proposal φ, we fuse its boundary probabilities and confidence scores by multiplication to generate the final confidence score p_f:

$$p_f = p^s_{t_s} \cdot p^e_{t_e} \cdot \sqrt{p_{cc} \cdot p_{cr}}. \tag{7}$$

Thus, we can get candidate proposals set $\Psi_p = \{\varphi_i = (t_s, t_e, p_f)\}_{i=1}^{N_p}$, where p_f is used for proposals retrieving during redundant proposals suppression.

Redundant Proposals Suppression. After generating candidate proposals, we need to remove redundant proposals to achieve higher recall with fewer proposals, where Non-maximum suppression (NMS) algorithm is widely used for this purpose. In BMN, we mainly adopt Soft-NMS algorithm[1], since it has proven its effectiveness in proposal generation task [18]. Soft-NMS algorithm suppresses redundant results via decaying their confidence scores. Soft-NMS generates suppressed final proposals set $\Psi'_p = \left\{\varphi_n = (t_s, t_e, p'_f)\right\}_{n=1}^{N'_p}$, where N'_p is the final proposals number. During experiment, we also try normal Greedy-NMS for fair comparison.

4. Experiments

4.1. Dataset and Setup

Dataset. We conduct experiments on two challenging datasets: **THUMOS-14** [15] dataset contains 413 temporal annotated untrimmed videos with 20 action categories; **ActivityNet-1.3** [5] is a large-scale action understanding dataset, containing action recognition, temporal detection, proposal generation and dense captioning tasks. ActivityNet-1.3 dataset contains 19994 temporal annotated untrimmed videos with 200 action categories, which are divided into training, validation and testing sets by ratio 2:1:1.

Implementation Details. For feature encoding, following previous works [18, 12], we adopt two-stream network [33] pre-trained on training set of ActivityNet-1.3, where spatial and temporal sub-networks adopt ResNet and BN-Inception network separately. The frame interval σ is set to 5 and 16 on THUMOS-14 and ActivityNet-1.3 separately.

On THUMOS-14, we set the length of observation window l_ω to 128 and the maximum duration length D to 64, which can cover length of 98% action instances. On ActivityNet, following [18, 20], we rescale each feature sequence to the length of the observation window $l_\omega = 100$ using linear interpolation, and the duration of corresponding annotations to range [0,1]. The maximum duration length D is set to 100, which can cover length of all action instances. To train BMN from scratch, we set learning rate to 0.001, batch size to 16 and epoch number to 10 for both datasets.

Table 2. Comparison between our method and other state-of-the-art temporal action proposal generation methods on validation set of ActivityNet-1.3 dataset in terms of AR@AN and AUC.

Method	[7]	[13]	[20]	[10]	[18]	BMN
AR@100 (val)	-	-	73.01	73.17	74.16	**75.01**
AUC (val)	59.58	63.12	64.40	65.72	66.17	**67.10**
AUC (test)	61.56	64.18	64.80	-	66.26	**67.19**

Table 3. Comparison between our method with state-of-the-art proposal generation methods SCNN [24], SST [3], TURN [12], TAG [36], CTAP [10], BSN [18] on THUMOS-14 dataset in terms of AR@AN, where SNMS stands for Soft-NMS.

Feature	Method	@50	@100	@200	@500	@1000
C3D	SCNN-prop	17.22	26.17	37.01	51.57	58.20
C3D	SST	19.90	28.36	37.90	51.58	60.27
C3D	TURN	19.63	27.96	38.34	53.52	60.75
C3D	BSN+NMS	27.19	35.38	43.61	53.77	59.50
C3D	BSN+SNMS	29.58	37.38	45.55	54.67	59.48
C3D	BMN+NMS	29.04	37.72	46.79	56.07	**60.96**
C3D	BMN+SNMS	**32.73**	**40.68**	**47.86**	**56.42**	60.44
2Stream	TAG	18.55	29.00	39.61	-	-
Flow	TURN	21.86	31.89	43.02	57.63	64.17
2Stream	CTAP	32.49	42.61	51.97	-	-
2Stream	BSN+NMS	35.41	43.55	52.23	61.35	65.10
2Stream	BSN+SNMS	37.46	46.06	53.21	60.64	64.52
2Stream	BMN+NMS	37.15	46.75	**54.84**	**62.19**	65.22
2Stream	BMN+SNMS	**39.36**	**47.72**	54.70	62.07	**65.49**

4.2. Temporal Action Proposal Generation

The goal of proposal generation task is to generate high quality proposals to cover action instances with high recall and high temporal overlap. To evaluate proposal quality, Average Recall (AR) under multiple IoU thresholds are calculated. Following conventions, IoU thresholds $[0.5 : 0.05 : 0.95]$ and $[0.5 : 0.05 : 1.0]$ are used for ActivityNet-1.3 and THUMOS-14 separately. We calculate AR under different Average Number of proposals (AN) as AR@AN, and calculate the Area under the AR vs. AN curve (AUC) as metrics on ActivityNet-1.3, where AN is varied from 0 to 100.

Table 4. Ablation comparison between BSN [18] and BMN in validation set of ActivityNet-1.3 in terms of AR@AN, AUC and inference speed. Inference speed here is the second (s) cost for processing a 3-minute video using a Nvidia 1080-Ti graphic card, including network inference time T_{inf}, proposal generation and proposal-feature generation (for BSN) time T_{pro} and the total inference time $T_{sum} = T_{inf} + T_{pro}$. *e2e* here means modules of network are trained jointly.

Method	Module	*e2e*	@100	AUC	T_{inf}	T_{pro}	T_{sum}
BSN	TEM	-	73.57	64.80	0.002	0.034	0.036
BSN	TEM+PEM	×	74.16	66.17	0.005	0.624	**0.629**
BMN	TEM	-	73.72	65.17	0.003	0.032	0.035
BMN	TEM+PEM	×	74.36	66.43	0.007	0.062	0.069
BMN	TEM+PEM	✓	75.01	67.10	0.005	0.047	**0.052**

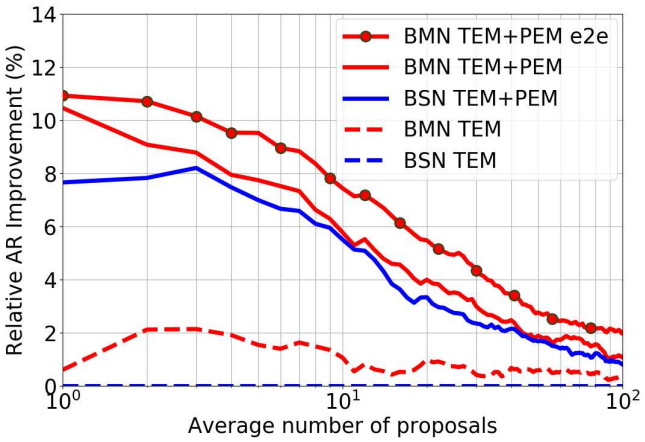

Figure 5. Ablation comparison between BSN and BMN in terms of relative AR improvement (%) vs AN curve on validation set of ActivityNet-1.3, where relative AR improvement is calculated based on BSN-TEM results.

Comparison with State-of-the-art Methods. Table 2 demonstrates the proposal generation performance comparison on validation and testing set of ActivityNet-1.3, where our method significantly outperforms other proposal generation methods. Especially, our method significantly improves AUC of validation set from 66.17% to 67.10% by 0.93%, which demonstrates that our method can achieve overall performance promotion.

Table 3 demonstrates the proposal generation performance comparison on testing set of THUMOS-14. Since different feature encoding methods and redundant proposal suppression methods can affect performance largely, following BSN [18], we adopt both C3D and two-stream feature, both normal Greedy-NMS and Soft-NMS for fair comparison. Experiment results suggest that (1) based on either C3D or two-stream feature, our method outperforms other methods significantly when proposal number varies from 10 to 1000; (2) no matter Greedy-NMS or Soft-NMS is adopted, our method outperforms other methods significantly; (3) Soft-NMS can improve average recall performance especially under small proposal number, which is helpful for temporal action proposal generation task. These results together suggest the effectiveness of our method and its effectiveness mainly due to its own architecture. Qualitative results are shown in Fig 6.

Table 5. Generalizability evaluation of BMN on validation set of ActivityNet-1.3 in terms of AR@AN and AUC.

	Seen		Unseen	
Training Data	AR@100	AUC	AR@100	AUC
Seen+Unseen	72.96	65.02	72.68	65.05
Seen	72.47	64.37	**72.46**	**64.47**

Ablation Comparison with BSN. To confirm the effect of the BM mechanism, we conduct more detailed ablation study and comparison of effectiveness and efficiency between BSN [18] and BMN. To achieve this, we evaluate the proposal quality and speed of BSN and BMN under multiple ablation configuration. The experiment results are shown in Table 4 and Fig 5, which demonstrate that:

1. Under similar network architecture and training objective, TEMs of BSN and BMN achieve similar proposal quality and inference speed, which provides a reliable comparison baseline;
2. Adding separately trained PEM, both BSN and BMN obtain significant performance promotion, suggesting that PEM plays an important role in the "local to global" proposal generation framework;
3. Jointly trained BMN achieves higher recall and faster speed than separately trained BMN, suggesting the effectiveness and efficiency of overall optimization;
4. Adding separately trained PEM, BMN achieves significant faster speed than BSN, since BM mechanism can directly generate confidence scores for all proposals simultaneously, rather than one-by-one respectively in BSN. Thus, PEM based on BM mechanism is more efficient than original PEM. Combining TEM and PEM jointly can further improve the efficiency.

Thus, these ablation comparison experiments suggest the effectiveness and efficiency of our proposed Boundary-Matching mechanism and unified BMN network, which can generate reliable confidence scores for all proposals simultaneously in fast speed.

Generalizability of Proposals. As a proposal generation method, an important property is the ability of generating high quality proposals for unseen action categories. To evaluate this property, following BSN [18], two un-overlapped action subsets: "Sports, Exercise, and Recreation" and "Socializing, Relaxing, and Leisure" of ActivityNet-1.3 are chosen as *seen* and *unseen* subsets separately. There are

Table 6. Action detection results on validation and testing set of ActivityNet-1.3, where our proposals are combined with video-level classification results generated by [37].

	validation				testing
Method	0.5	0.75	0.95	Average	Average
CDC [23]	43.83	25.88	0.21	22.77	22.90
SSN [34]	39.12	23.48	5.49	23.98	28.28
Lin et al. [20]	44.39	29.65	7.09	29.17	32.26
BSN [18] + [37]	46.45	29.96	8.02	30.03	32.87
Ours + [37]	**50.07**	**34.78**	**8.29**	**33.85**	**36.42**

Table 7. Action detection results on testing set of THUMOS14, where video-level classifier UntrimmedNet [31] and proposal-level classifier SCNN-Classifier [24] are combined with proposals.

Method	classifier	0.7	0.6	0.5	0.4	0.3
SST [3]	SCNN-cls	-	-	23.0	-	-
TURN[12]	SCNN-cls	7.7	14.6	25.6	33.2	44.1
BSN [18]	SCNN-cls	15.0	22.4	29.4	36.6	43.1
Ours	SCNN-cls	**17.0**	**24.5**	**32.2**	**40.2**	**45.7**
SST [3]	UNet	4.7	10.9	20.0	31.5	41.2
TURN[12]	UNet	6.3	14.1	24.5	35.3	46.3
BSN [18]	UNet	20.0	28.4	36.9	45.0	53.5
Ours	UNet	**20.5**	**29.7**	**38.8**	**47.4**	**56.0**

87 and 38 action categories, 4455 and 1903 training videos, 2198 and 896 validation videos on *seen* and *unseen* subsets separately. And we adopt C3D network [28] pre-trained on Sports-1M dataset [16] for feature extraction, to guarantee the validity of experiments. We train BMN with *seen* and *seen+unseen* training videos separately, and evaluate both BMN models on *seen* and *unseen* validation videos separately. Results in Table 5 demonstrate that the performance drop is very slight in unseen categories, suggesting that BMN achieves great generalizability to generate high quality proposals for unseen actions, and can learn a general concept of when an action may occur.

4.3. Action Detection with Our Proposals

Another important aspect of evaluating the proposal quality is to put proposals in temporal action detection framework and evaluate its detection performance. Mean Average Precision (mAP) is adopted as the evaluation metric of temporal action detection task, where we calculate Average Precision (AP) on each action category respectively. mAP with IoU thresholds $\{0.5, 0.75, 0.95\}$ and average mAP with IoU thresholds $[0.5 : 0.05 : 0.95]$ are used on ActivityNet-1.3, while mAP with IoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ are used on THUMOS-14.

To achieve this, we adopt the two-stage "detection by classifying proposals" temporal action detection framework to combine BMN proposals with state-of-the-art action classifiers. Following BSN [18], on ActivityNet-1.3, we adopt top-1 video-level classification results generated by method [37] and use confidence scores of BMN proposals for detection results retrieving. On THUMOS-14, we use both top-2 video-level classification results generated by UntrimmedNet [31], and proposal-level SCNN-classifier to generate classification result for each proposal. For ActivityNet-1.3 and THUMOS-14 datasets, we use first 100 and 200 temporal proposals per video separately.

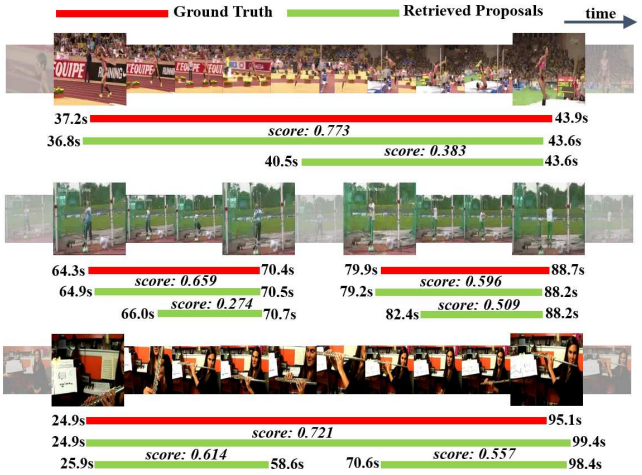

Figure 6. Visualization examples of proposals and BM map generated by BMN on THUMOS-14 and ActivityNet-1.3 dataset.

The experiment results on ActivityNet-1.3 are shown in Table 6, which demonstrate that BMN proposals based detection framework significantly outperform other state-of-the-art temporal action detection methods. The experiment results on THUMOS-14 are shown in Table 7, which suggest that: (1) no matter video-level or proposal-level action classifier is used, our method achieves better detection performance than other state-of-the-art proposal generation methods; (2) using BMN proposals, video-level classifier [31] achieves significant better performance than proposal-level classifier [24], indicating that BMN can generate confidence scores reliable enough for retrieving results.

5. Conclusion

In this paper, we introduced the Boundary-Matching mechanism for evaluating confidence scores of densely distributed proposals, which is achieved via denoting proposal as BM pair and combining all proposals as BM confidence map. Meanwhile, we proposed the Boundary-Matching Network (BMN) for effective and efficient temporal action proposal generation, where BMN generates proposals with precise boundaries and flexible duration via combining high probability boundaries, and simultaneously generates reliable confidence scores for all proposals based on BM mechanism. Extensive experiments demonstrate that BMN outperforms other state-of-the-art proposal generation methods in both proposal generation and temporal action detection tasks, with remarkable efficiency and generalizability.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nmsimproving object detection with one line of code. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5562–5570. IEEE, 2017. 6

[2] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *Proceedings of the British Machine Vision Conference*, 2017. 2

[3] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6373–6382. IEEE, 2017. 1, 2, 3, 6, 8

[4] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016. 1, 2

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1, 6

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 2

[7] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5727–5736. IEEE, 2017. 6

[8] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016. 1, 2, 3

[9] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 2

[10] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–83, 2018. 6

[11] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. In *Proceedings of the British Machine Vision Conference*, 2017. 3

[12] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3648–3656. IEEE, 2017. 2, 3, 6, 8

[13] Bernard Ghanem, Juan Carlos Niebles, Cees Snoek, Fabian Caba Heilbron, Humam Alwassel, Ranjay Khrisna, Victor Escorcia, Kenji Hata, and Shyamal Buch. Activitynet challenge 2017 summary. *CVPR ActivityNet Workshop*, 2017. 6

[14] Yupan Huang, Qi Dai, and Yutong Lu. Decoupling localization and classification in single shot temporal action detection. *arXiv preprint arXiv:1904.07442*, 2019. 2

[15] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 1, 6

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 8

[17] Zhengfa Liang, Yiliu Feng, Yulan Guo, Hengzhu Liu, Linbo Qiao, Wei Chen, Li Zhou, and Jianfeng Zhang. Learning deep correspondence through prior and posterior feature constancy. *arXiv preprint arXiv:1712.01039*, 7(8), 2017. 2

[18] Tianwei Lin, Xu Zhao, and SU Haisheng. Bsn: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[19] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 988–996. ACM, 2017. 2, 3

[20] Tianwei Lin, Xu Zhao, and Zheng Shou. Temporal convolution based action proposal: Submission to activitynet 2017. *CVPR ActivityNet Workshop*, 2017. 6, 8

[21] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 2

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

[23] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1417–1426. IEEE, 2017. 8

[24] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016. 1, 2, 6, 8

[25] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2, 3

[26] Xiao Song, Xu Zhao, Hanwen Hu, and Liangji Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. *arXiv preprint arXiv:1803.05196*, 2018. 2

[27] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with

3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[28] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017. 8

[29] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176. IEEE, 2011. 2

[30] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 2

[31] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017. 8

[32] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *CoRR*, abs/1507.02159, 2015. 2

[33] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 6

[34] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017. 8

[35] Runhao Zeng, Chuang Gan, Peihao Chen, Wenbing Huang, Qingyao Wu, and Mingkui Tan. Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization. *IEEE Transactions on Image Processing*, 2019. 2

[36] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2933–2942. IEEE, 2017. 2, 3, 6

[37] Y Zhao, B Zhang, Z Wu, S Yang, L Zhou, S Yan, L Wang, Y Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 2017. 8