BACNET: BOUNDARY-ANCHOR COMPLEMENTARY NETWORK FOR TEMPORAL ACTION DETECTION

*Zixuan Zhao, Dongqi Wang and Xu Zhao**

Department of Automation, Shanghai Jiao Tong University, China

{zhaozixuan, wangdq0124, zhaoxu}@sjtu.edu.cn

2022 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-8563-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICME52920.2022.9859756

ABSTRACT

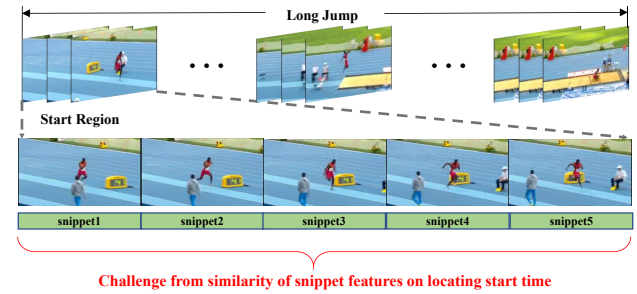
The task of temporal action detection aims to locate and classify action segments in untrimmed videos. Most existing works usually consist of two components: snippet-level boundary segmentation and anchor-level action evaluation. These two components, however, are typically designed irrelevantly, so the detection accuracy is undermined due to vague boundaries and complex video content. To tackle this problem, we design two supplementary modules. One module, termed as Anchor Aware Module (AAM), uses temporal and semantic related anchors to enhance snippet feature. The other module, named Boundary Aware Module (BAM), endows anchor feature with structured representation using intermediate supervision. Moreover, the ConvLSTM is applied to establish temporal relation in BAM with the structured representation. These two modules are integrated as the Boundary-Anchor Complementary Network (BACNet), which achieves the state-of-the-art performance on both THUMOS-14 and ActivityNet-1.3 datasets.

***Index Terms*—** Video Understanding, Temporal Action Detection, Complementary Network

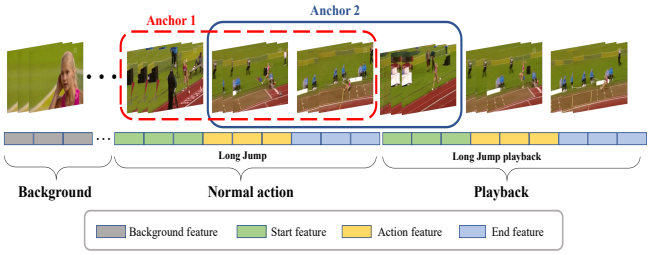
1. INTRODUCTION

Temporal Action Detection (TAD) is a fundamental task in video understanding, which aims to locate the action boundaries and classify the action segments in untrimmed video. Most existing methods contain two branches: a boundary segmentation branch, designed for boundary location with snippet-level feature; an action evaluation branch, designed for confidence evaluation with anchor-level feature. However, the interaction between these two branches is usually ignored, which brings two drawbacks: (1) Confused boundary probability; (2) Inaccurate anchor confidence.

In terms of boundary segmentation branch, each snippet is classified as start or end with certain probabilities to locate the boundaries in a video. But, boundaries of some actions are too vague to tell from only surrounding snippets, which will inhibit boundary location. As shown in Fig.1(a), considering the similarity between snippets in the start period of



(a) Confusing start time in the start period.



(b) Similar anchor content without structured representation.

Fig. 1. (a) In the start region, only based on the similar local features, model cannot accurately identify the start time. (b) The dotted box (anchor 1) contains complete *Long Jump* action with reasonable structure (start region, action region and end region) and the solid box (anchor 2) represents an illogical action with disordered structure. However, they have similar content when ignoring the structured information.

Long Jump, it is difficult to distinguish which snippet is the real start. If the model can see the whole *Long Jump* action, boundary location might be easier. In order to alleviate this problem, traditional convolution [1, 2], graph convolution [3] and self-attention [4, 5] are used to expand the snippet horizons. However, these methods still focus on snippet-level feature for contextual relation construction, ignoring the segment context between boundaries. Thus, inadequate snippet feature will produce inaccurate boundary probabilities.

In terms of action evaluation branch, it is used to estimate the confidence of each anchor. The anchor is composed of several snippets to represent a video segment. Most meth-

*Corresponding author. This work has been funded in part by the NSFC grants 62176156 and Shanghai Engineering Research Center of Intelligent Control and Managemen.

ods [1, 6] sample features within the anchor uniformly, which make anchor representation lack boundary information and temporal relation. Taking the sports video in Fig.1(b) as an example, anchor 1 represents a real action with rational temporal relation (start region, action region and end region). Anchor 2 represents an illogical action with disordered temporal relation (action region and end region from normal action, but start region from the playback), which has similar content with anchor 1. If the model could not establish the structured representation in anchors, it leads to similar high confidence scores for both anchor 1 and anchor 2.

Based upon the problems above, we propose a novel network named Boundary-Anchor Complementary Network, which consists of two crucial modules. (1) In order to enable anchor feature to integrate structured representation, Boundary Aware Module (BAM) is designed. We contend that the features of different regions inside an anchor have different meanings (*i.e.*, start feature, action feature, end feature), which are defined as structured features. As a result of it, to emphasize the boundary information in an anchor, BAM applies boundary supervision to construct the structured features. Furthermore, for reasonable anchor representation, ConvLSTM is used to establish the multi-level temporal relations among structured features. (2) In order to expand the receptive field of boundary snippet, Anchor Aware Module (AAM) is proposed. AAM builds temporal and semantic relations between snippets and anchors, which provides boundary with not only local features nearby boundary but also segment contexts between boundaries. In summary, our work has the following contributions:

1. To our knowledge, this is the first work attempting to build the complementary relation between boundary segmentation and action evaluation. We exploit the complementary information between each other to enhance the feature within a TAD framework.
2. We introduce two novel modules. Boundary Aware Module establishes the structured feature representation of anchors. Anchor Aware Module models temporal and semantic related anchor contexts, which are applied to enhance the boundary feature.
3. Boundary-Anchor Complementary Network obtains superior performance on two popular datasets, *i.e.*, THUMOS-14 and ActivityNet-1.3.

2. RELATED WORK

Temporal Action Detection. (1) Top-down method: top-down method defines anchors on the video or feature pyramid sequence to produce proposals. Most works [7, 8, 9] use anchor-level feature to predict the confidence score of each anchor. (2) Bottom-up method: bottom-up method [10, 11, 12] predicts the start score, end score, and actionness score of snippets, which can be combined to generate action proposals. (3) Combined method: combined method [1, 2, 13] integrates these two methods for better performance. These works predict boundaries and anchor confidences separately, which do not consider the interaction between each part. In contrast, an intermediate bridge considering these two parts as a whole, is built in this paper.

Feature representation. (1) To enrich the snippet feature, many works contribute to expanding the receptive field of snippets or enriching context with other snippets. In [3], it is believed that adjacent snippets and semantically similar snippets provide context properties and aggregates snippet-level features with graph convolution network. TCANet [14] encodes the local and global context features with a channel grouping strategy. However, these works merely concentrate on local features, lacking segment contexts. (2) To enhance the representation of anchors, some works establish relations between different anchors. PGCN [15] establishes graph structure between proposals. ContextLoc [16] designs a multi-level feature model, which explores query-and-retrieval process to enrich local and global context simultaneously. However, these models still depend on proposals generated by other methods. They enrich the proposal features while not paying attention to feature representation within the proposal. In this work, we address the challenges by building two complementary modules to let snippet feature has segment contexts and anchor feature has structured representation.

3. METHOD

3.1. Overview

Problem Definition. The untrimmed video is composed of l_f frames. Every σ frames are regarded as a snippet. And then, the video can be divided into T snippets with stride d, where d is defined as distance between snippet centers. Video snippets are encoded into feature sequence $F_{in} \in \mathbb{R}^{C' \times T}$ via action recognition methods, where C' is the dimension of feature. The annotations of untrimmed video are action instances $\{\psi_i | \psi_i = (t_{i,s}, t_{i,e}, c_i)\}$, where $t_{i,s}, t_{i,e}$ and c_i are start time, end time and action category, respectively.

Network Architecture. The architecture of BACNet is shown in Fig. 2. BACNet is mainly composed of four modules: Boundary Aware Module (BAM), Anchor Aware Module (AAM), Boundary Segmentation Module (BSM) and Action Evaluation Module (AEM). The design of BSM and AEM follows the prior work[1].

Firstly, the video feature sequence F_{in} is fed into a basic block stacked by two 1D convolution layers, which is used to produce base feature $F \in \mathbb{R}^{C \times T}$. Next, the base feature F is simultaneously fed into BSM and BAM. BSM is applied to generate start/end features $F_s, F_e \in \mathbb{R}^{C \times T}$ and start/end probabilities $P_s = \{p_s^i\}_{i=1}^T, P_e = \{p_e^i\}_{i=1}^T$ by 1D convolution. BAM aims to generate 2D anchor representation map $F_A = \{f_A^{c,j,i}\} \in \mathbb{R}^{C \times D \times T}$ with continuous start time and temporal duration. $f_A^{c,j,i}$ represents the anchor feature with the start time i and duration j, and D is predefined maximum anchor duration. After generating 2D anchor representation map, AEM is used to produce two confidence maps

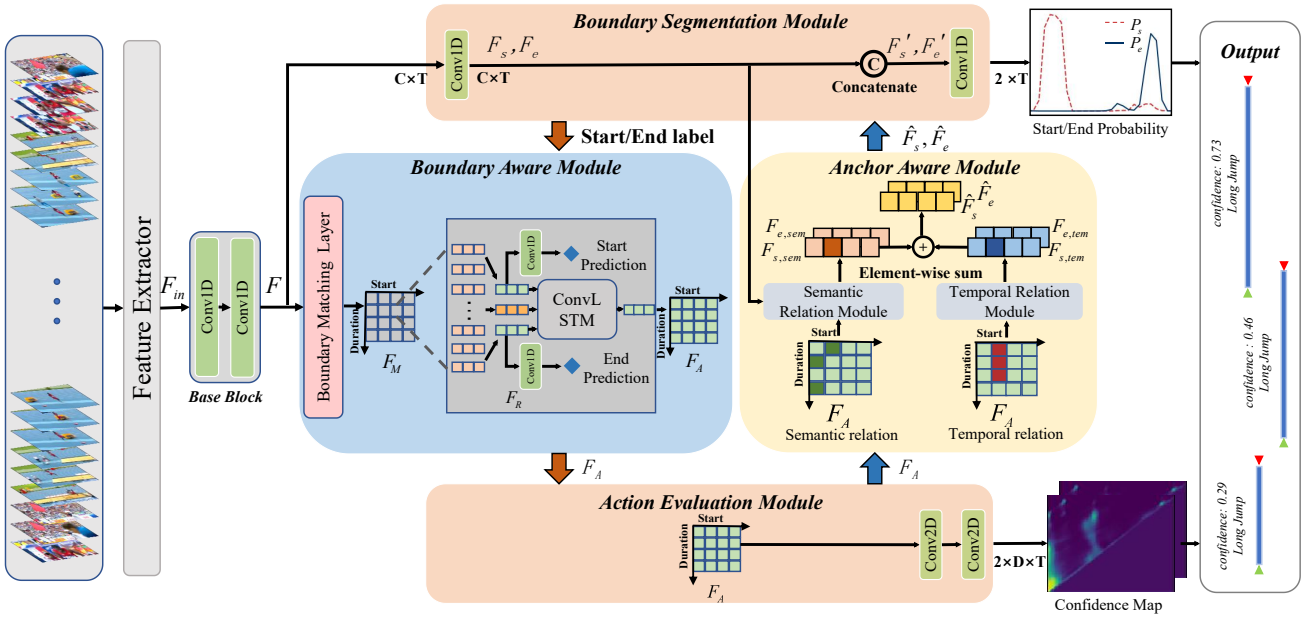


Fig. 2. Overview of our BACNet. We extract the feature from untrimmed video and re-scale it to a fixed length. BSM and AEM are used to generate boundary probabilities for each snippet and evaluate anchor confidence respectively. The BAM and AAM are adopted to exchange information between BSM and AEM. Finally, we use the boundary probabilities and anchor confidence maps to generate proposals.

$M_{cls} \in \mathbb{R}^{D\times T}, M_{reg} \in \mathbb{R}^{D\times T}$ by 2D convolution. Finally, to refine boundary representation in BSM, AAM is designed to construct enhanced start/end features $\hat{F}_s \in \mathbb{R}^{C\times T}$ and $\hat{F}_e \in \mathbb{R}^{C\times T}$. BAM and AAM will be detailed later.

3.2. Boundary Aware Module

To get a reliable confidence map, the representation of anchors must be more discriminating. Boundary Aware Module establishes structured anchor representation, which contains two vital stages: structured feature construction and the structured feature utilization.

Structured feature construction. BAM adopts Boundary-Matching Layer [1] to generate the original anchor feature map $F_M \in \mathbb{R}^{C\times N\times D\times T}$. Video segments with arbitrary length could be represented by an anchor in the map. The anchor feature denoted as $f_M^{j,i} \in \mathbb{R}^{C\times N}$ is uniformly sampled from start t_i to end t_{i+j}, where N is the sampling number.

To construct structured feature $F_R \in \mathbb{R}^{C\times R\times D\times T}$, BAM conducts 3D convolution layer to reduce $f_M^{j,i}$ dimension length from N to R, which can be regarded as R regions. Next, to highlight boundary information in $f_R^{j,i} \in \mathbb{R}^{C\times R}$, the first and last region features are fed into a 1D convolution layer to generate anchor boundary probabilities $P_{s,map}, P_{e,map}$.

Structured feature utilization. For final anchor representation F_A, recent works aggregate snippet features with pooling [16] or convolution operation [1], which treat different snippet features equally without temporal relation. Obviously, multi-level temporal relations exist in F_R. Intra relation: the features inside $f_M^{j,i}$ have temporal order. Inter relation: in a macro view, each anchor in the F_R has similar content with its neighbors, which are locally related. For example, an anchor from start t_i to end t_j contains similar content with its neighbour anchor, which is from start t_{i-1} to end t_j.

In order to handle these two relations simultaneously, ConvLSTM [17] is applied on F_R to get a stronger anchor representation map F_A, as formulated in Eq. 1, where '$\cdot||\cdot$' is concatenate operation, '$\circ$' is the Hadamard product, '$*$' is the convolution operator, 'W' and 'b' are learnable parameters. 'IG', 'FG', 'OG', 'CG' denote input gate, forget gate, output gate and cell gate, respectively. $f_R^t \in \mathbb{R}^{C\times D\times T}$ represents the t-th region feature in anchors. 'H^t' is hidden state whose shape is the same as f_R^t. F_A is the last output of H^t.

$$\begin{aligned} IG^t &= Relu(W_{input} * (f_R^t || H^{t-1}) + b_{input}) \\ FG^t &= Relu(W_{forget} * (f_R^t || H^{t-1}) + b_{forget}) \\ OG^t &= Relu(W_{output} * (f_R^t || H^{t-1}) + b_{output}) \\ CG^t &= Tanh(W_{cell} * (f_R^t || H^{t-1}) + b_{cell}) \\ Cell^t &= (FG^t \circ Cell^{t-1}) + (IG^t \circ CG^t) \\ H^t &= OG^t \circ Tanh(Cell^t) \end{aligned} \tag{1}$$

3.3. Anchor Aware Module

As discussed before, taking into account segment features can provide supportive cues to discriminate boundaries. Anchor Aware Module is designed to fully exploit temporal and semantic relations between boundary features and anchor features. The structure of AAM is illustrated in Fig. 2, which is composed of Temporal Relation Module (TRM) and Semantic Relation Module (SRM).

TRM. TRM is used to expand boundary horizons with temporal related anchors. As shown in Fig. 2, red anchors con-

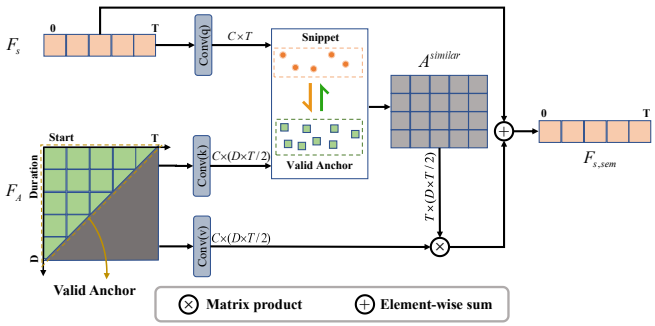


Fig. 3. Illustration of SRM. This module constructs the semantic relation between boundary snippets and valid area in anchor map.

taining segment information have the same start time t_i with the dark blue snippet, which may help start classification at time t_i. In order to grasp the salient segment feature, TRM takes MaxPooling among the anchors with the same start time at each moment, which generates temporal related feature $F_{s,tem} \in \mathbb{R}^{C\times T}$. As for $F_{e,tem} \in \mathbb{R}^{C\times T}$, TRM takes MaxPooling among the anchors, which have the same end time.

SRM. In addition to temporal related anchors, anchors that are semantic related to boundary features will also help boundary segmentation. Inspired by [18], SRM models semantic relation between F_A and F_s, F_e. Taking the F_s in Fig. 3 as an example, SRM uses 1D convolution to transform F_s to the query $F_{s,query} \in \mathbb{R}^{C\times T}$, and the valid area (left upper triangular matrix) of F_A to key $F_{A,key} \in \mathbb{R}^{C\times(D\times T/2)}$ and value $F_{A,value} \in \mathbb{R}^{C\times(D\times T/2)}$. Furthermore, SRM establishes a weight map $A^{similar} \in \mathbb{R}^{T\times(D\times T/2)}$ between $F_{A,key}$ and $F_{s,query}$ to retrieve the related value $F_{A,value}$. The $A^{similar}$ is calculated as:

$$A^{similar} = f_{norm}({F_{s,query}}^T \times F_{A,key}) \tag{2}$$

where f_{norm} means Softmax operation. After that, the semantic related feature $F_{s,sem}$ is obtained by:

$$F_{s,sem} = F_{A,value} \times {A^{similar}}^T + F_s \tag{3}$$

The final enhanced feature $\hat{F}_s$ is obtained by temporal related feature $F_{s,tem}$ and semantic related feature $F_{s,sem}$ as Eq. 4. Just like $\hat{F}_s$, $\hat{F}_e$ is obtained in the same way.

$$\hat{F}_s = F_{s,tem} + F_{s,sem} \tag{4}$$

By concatenating $\hat{F}_s$ and F_s, $\hat{F}_e$ and F_e, the updated boundary features F'_s, F'_e contain not only local information but also segment contexts.

3.4. Training

Label Assignment. In order to expand the boundary from moment to region, we define the start region and end region as $r_s = [t_s - 1.5d, t_s + 1.5d]$ and $r_e = [t_e - 1.5d, t_e + 1.5d]$. We calculate the overlap between snippet and start/end regions to get the probabilities $G_s = \{g_s^i\}_{i=1}^T$ and $G_e = \{g_e^j\}_{j=1}^T$ as start/end labels. For anchor map, the ground truth $G_{map} = \{g_{map}^{j,i}\}$ is generated by calculating the Intersection over Union (IoU) between each anchor area and action area.

Loss in BSM. For the Boundary Segmentation Module, we can obtain boundary probability sequences P_s, P_e. We use L_B to represent binary logistic loss, which is formulated as Eq. 5. The boundary segmentation loss is defined as Eq. 6.

$$L_B = \frac{1}{T}\sum_{i=1}^{T}(\alpha \cdot g^i \cdot log(p^i) + \beta \cdot (1-g^i) \cdot log((1-p^i))) \tag{5}$$

$$L_{boundary} = L_B(P_s, G_s) + L_B(P_e, G_e) \tag{6}$$

g^i is converted by function $sign(g^i - 0.5)$ from $[0,1]$ to $\{0,1\}$. Denoting $T^+ = \sum sign(g^i - 0.5)$ and $T^- = \sum sign(0.5 - g^i)$, the α and β are defined as $\frac{T}{T^+}$ and $\frac{T}{T^-}$.

Loss in AEM. For the Action Evaluation Module, we can obtain anchor classification map M_{cls} and anchor regression maps M_{reg}. Same as L_B, L_{Mcls} represents binary logistic loss and L_{Mreg} represents L2 loss. The anchor evaluation loss is defined as:

$$L_{map} = L_{Mcls}(M_{cls}, G_{map}) + L_{Mreg}(M_{reg}, G_{map}) \tag{7}$$

For calculating L_{Mcls}, signal function $sign(g_{map}^{j,i} - 0.9)$ is used to convert $g_{map}^{j,i}$ from $[0,1]$ to $\{0,1\}$.

Loss in BAM. In Boundary Aware Module, we can get anchor boundary probabilities $P_{s,map}, P_{e,map}$ from each anchor. Especially, we add anchor boundary classification loss L_{mb} as intermediate supervision (Eq. 8), where $G_{s,map} = \{g_{s,map}^{j,i}\}$, $G_{e,map} = \{g_{e,map}^{j,i}\}$. For each anchor, the $g_{s,map}^{j,i}$ equals to g_s^i and the $g_{e,map}^{j,i}$ equals to g_e^{i+j}. And then, $G_{s,map}$ and $G_{e,map}$ are flattened to vectors.

$$L_{mb} = L_B(P_{s,map}, G_{s,map}) + L_B(P_{e,map}, G_{e,map}) \tag{8}$$

Total loss. The total loss of the BACNet includes boundary segmentation loss, anchor evaluation loss and anchor boundary classification loss:

$$L = L_{boundary} + L_{map} + L_{mb} + \lambda_\theta L_2(\Theta) \tag{9}$$

where $L_2(\Theta)$ represents L_2 regularization term and λ_θ represents weight decay, which is set to 10^{-4}.

3.5. Inference

We can obtain start/end probabilities P_s, P_e from BSM, and confidence maps M_{cls}, M_{reg} from AEM. Following [1], the boundary points p_s^i and p_e^{i+j} are combined into proposals when they are local peak or their values are greater than a threshold. Then, an action recognition model is used to generate classification score p_{class} for each proposal. The final confidence score is defined as Eq. 10.

$$S_{final} = p_s^i \cdot p_e^{i+j} \cdot M_{cls}^{(j,i)} \cdot M_{reg}^{(j,i)} \cdot p_{class} \tag{10}$$

4. EXPERIMENTS

4.1. Dataset Settings

We evaluate BACNet on two challenging datasets THUMOS-14 and ActivityNet-1.3 [19]. THUMOS-14 has 20 labeled action categories with temporal annotations, which consists of 200 validation videos and 213 testing videos. An action recognition model is used to extract feature sequence from

untrimmed video. On THUMOS-14, we use sliding window with the length T of 128 to cut the whole video feature sequence into several windows. And the maximum anchor duration D is set as 64, which can cover 98% action instances. ActivityNet-1.3 contains 19994 temporally annotated videos with 200 action categories. For each video, we resize the feature sequence length T to 100 using linear interpolation, and the D is set to 100. BACNet is trained with batch size of 16 and training epoch of 10. After running 5 epochs on THUMOS-14 and 7 epochs on ActivityNet-1.3, we use step scheduler to reduce the learning rate from 10^{-3} to 10^{-4} .

4.2. Comparison with State-of-the-art Methods

The input feature sequence is provided by [3] from pre-trained two-stream network [20]. Table 1 shows the performance comparision between BACNet and other state-of-the-art models on THUMOS-14, which reports the *mAP* at tIoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ and average *mAP*. Table 2 shows the superior performance of BACNet on ActivityNet-1.3. We report the *mAP* at tIoU thresholds $\{0.5, 0.75, 0.95\}$ and average *mAP*. Specifically, the average *mAP* is calculated of tIoU threshold between 0.5 and 0.95 with the step of 0.05.

On THUMOS-14, BACNet obtains outstanding performance. Compared with the previous best method [21], BACNet gains 2.38% improvement on average *mAP*. More specifically, the performance of BACNet at all thresholds surpasses other methods, which demonstrates the effectiveness of our method. On ActivityNet-1.3, in terms of *mAP*@0.75 and the average *mAP*, BACNet outperforms all other existing methods, which verifies the predictions of BACNet are more reliable generally. However, the improvement on ActivityNet-1.3 is less than the improvement on THUMOS-14. The main reasons may lie in two aspects. Firstly, the performance of BACNet relies on the accuracy of boundaries, but in ActivityNet-1.3, there are many daily actions whose boundaries are vague, which undermines the performance of this dataset. Secondly, for fair comparisons with other models, we choose the offline feature as the input of BACNet. However, the feature dimension of ActivityNet-1.3 is smaller than the dimension of THUMOS-14, which means the feature expression of ActivityNet-1.3 is weaker than THUMOS-14.

4.3. Ablation Study

To fully explore the contribution of each component in BACNet, extensive ablation studies are conducted.

Total Framework. In order to further verify the effectiveness of complementary information in BACNet, we add BAM and AAM on the base model individually.

According to Table 3, we find the performance can be improved by BAM and AAM individually. Average *mAP* is improved by 3.10% when adding BAM alone, which verifies the structured features of anchors produce more precise anchor confidence. Average *mAP* is improved by 3.54% when adding AAM alone, which validates AAM brings supplementary contexts from anchors for better boundary judgement. And the best performance appears when two modules are combined, which shows these two modules do not conflict. In general, experiments indicate that information exchange between boundaries and anchors is necessary.

Table 1. The performance comparison with state-of-art methods on THUMOS-14, where the classification results are generated by [22]. Bold text indicates the best results.

Method	mAP@tIoU (%)					
	0.3	0.4	0.5	0.6	0.7	Avg.
BSN [10]	53.50	45.00	36.90	28.40	20.00	36.76
BMN [1]	56.00	47.40	38.80	29.70	20.50	38.48
DBG [13]	57.80	49.40	39.80	30.20	21.70	39.78
PGCN [15]	63.60	57.80	49.10	-	-	34.10
GTAD [3]	54.50	47.60	40.20	30.80	23.40	39.30
TSI [6]	61.00	52.10	42.60	33.20	22.40	42.26
BSN++ [2]	59.90	49.50	41.30	31.90	22.80	41.08
RTD-Net [23]	68.30	62.30	51.90	38.80	23.70	49.00
ContextLoc [16]	68.30	63.80	54.30	41.80	26.20	50.88
Anchor-free [21]	67.30	62.40	55.50	43.70	31.10	52.00
BACNet (ours)	**69.62**	**64.16**	**56.35**	**46.54**	**35.21**	**54.38**

Table 2. The performance comparison with state-of-art methods on ActivityNet-1.3, where the classification results are generated by [24]. Bold text indicates the best results.

Method	mAP@tIoU (%)			
	0.5	0.75	0.95	Avg.
BSN [10]	46.45	29.96	8.02	30.03
BMN [1]	50.07	34.78	8.29	33.85
PGCN [15]	48.26	33.16	3.27	31.11
GTAD [3]	50.36	34.60	**9.02**	34.09
TSI [6]	51.18	35.02	6.59	34.15
BSN++ [2]	51.27	35.70	8.33	34.88
RTD-Net [23]	47.21	30.68	8.61	30.83
ContextLoc [16]	**56.01**	35.19	3.55	34.23
Anchor-free [21]	52.40	35.30	6.50	34.40
BACNet (ours)	51.68	**36.06**	6.83	**34.90**

Table 3. Ablation study on the total framework of BACNet. *mAP@0.5* and average *mAP* are reported on THUMOS-14.

Method	mAP@0.5	Avg. mAP
base	51.97	49.31
base + BAM	54.21	52.41
base + AAM	54.52	52.85
BACNet	**56.35**	**54.38**

BAM. To explore the contributions of structured feature construction and utilization of anchors, we ablate the boundary supervision and ConvLSTM respectively. As shown in Table 4, "base + Boundary Supervision" represents anchor features with boundary supervision, while applies traditional convolution to aggregate the features. "base + ConvLSTM" uses ConvLSTM to construct temporal relation with original sampled features.

From the comparison of "base", "base + Boundary Supervision" and "base + BAM", when we only build the structured

Table 4. Ablation study on BAM. *mAP@0.5* and average *mAP* are reported on THUMOS-14.

Method	mAP@0.5	Avg. mAP
base	51.97	49.31
base + Boundary Supervision	51.99	50.17
base + ConvLSTM	53.03	51.08
base + BAM	**54.21**	**52.41**

Table 5. Ablation study on AAM. *mAP@0.5* and average *mAP* are reported on THUMOS-14.

Method	mAP@0.5	Avg. mAP
base	51.97	49.31
base + Temporal Relation	53.65	51.82
base + Semantic Relation	51.05	49.65
base + AAM	**54.52**	**52.85**

features, average *mAP* is improved by 0.86%. Moreover, when ConvLSTM is applied to construct temporal relation, the performance is improved from 50.17% to 52.41%, which promotes the performance greatly. The promotion illustrates that constructing structured features is very important for the anchor representation. Meanwhile, compared with traditional convolution, ConvLSTM can make better use of the structured features, which indicates that multi-level temporal relations among anchors are significant.

AAM. To explore the impacts of temporal and semantic anchor features, we ablate the TRM and SRM in AAM. As shown in Table 5, "base + Temporal Relation" and "base + Semantic Relation" options indicate whether the model is equipped with Temporal Relation Module and Semantic Relation Module respectively.

When comparing "base + Temporal Relation", "base + Semantic Relation" and "base + AAM", we have a clear understanding of the effectiveness of the anchor contexts. The promotion is limited when we merely add Semantic Relation Module, but the performance of "base + AAM" is improved significantly, which illustrates the temporal and semantic related contexts play a mutually reinforcing role in AAM.

5. CONCLUSION

In this paper, we concentrate on the complementary information between boundary segmentation and action evaluation in TAD task. To produce the discriminating anchor representation, we design Boundary Aware Module to construct structured representation of anchors with boundary information, which highlights the boundary while constructing the multi-level temporal relations within anchors. To expand the receptive field of snippets with segment information, Anchor Aware Module is proposed to enrich local feature with temporal and semantic related anchor features. These two modules are integrated into one framework named BACNet. Experiments show that the BACNet achieves superior performance.

6. REFERENCES

[1] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *ICCV*, 2019.

[2] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan, "Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation," in *AAAI*, 2021.

[3] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *CVPR*, 2020.

[4] Xin Qin, Hanbin Zhao, Guangchen Lin, Hao Zeng, Songcen Xu, and Xi Li, "Pcmnet: Position-sensitive context modeling network for temporal action localization," *arXiv*, 2021.

[5] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu, "Class semantics-based attention for action detection," in *ICCV*, 2021.

[6] Shuming Liu, Xu Zhao, Haisheng Su, and Zhilan Hu, "Tsi: Temporal scale invariant network for action proposal generation," in *ACCV*, 2020.

[7] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen, "Temporal context network for activity localization in videos," in *ICCV*, 2017.

[8] Tianwei Lin, Xu Zhao, and Zheng Shou, "Single shot temporal action detection," in *ACMMM*, 2017.

[9] Zheng Shou, Dongang Wang, and Shih-Fu Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016.

[10] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *ECCV*, 2018.

[11] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin, "Temporal action detection with structured segment networks," in *ICCV*, 2017.

[12] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei, "Gaussian temporal awareness networks for action localization," in *CVPR*, 2019.

[13] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji, "Fast learning of temporal action proposal via dense boundary generator," in *AAAI*, 2020.

[14] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang, "Temporal context aggregation network for temporal action proposal refinement," in *CVPR*, 2021.

[15] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.

[16] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua, "Enriching local and global contexts for temporal action localization," in *ICCV*, 2021.

[17] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015.

[18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *CVPR*, 2018.

[19] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015.

[20] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks for action recognition in videos," *PAMI*, 2018.

[21] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu, "Learning salient boundary feature for anchor-free temporal action localization," in *CVPR*, 2021.

[22] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017.

[23] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu, "Relaxed transformer decoders for direct action proposal generation," *arXiv preprint arXiv:2102.01894*, 2021.

[24] Yue Zhao, Bowen Zhang, Zhirong Wu, Shuo Yang, Lei Zhou, Sijie Yan, Limin Wang, Yuanjun Xiong, D Lin, Y Qiao, et al., "Cuhk & ethz & siat submission to activitynet challenge 2017," *arXiv*, 2017.