Estimation of Reliable Proposal Quality for Temporal Action Detection

Junshan Hu*
junshan@mail.ustc.edu.cn
University of Science and Technology of China

Chaoxu Guo
chaoxu.gcx@alibaba-inc.com
Alibaba Group

Liansheng Zhuang†
lszhuang@ustc.edu.cn
University of Science and Technology of China

Biao Wang
eric.wb@alibaba-inc.com
Alibaba Group

Tiezheng Ge
tiezheng.gtz@alibaba-inc.com
Alibaba Group

Yuning Jiang
mengzhu.jyn@alibaba-inc.com
Alibaba Group

Houqiang Li
lihq@ustc.edu.cn
University of Science and Technology of China

ABSTRACT

Temporal action detection (TAD) aims to locate and recognize the actions in an untrimmed video. Anchor-free methods have made remarkable progress which mainly formulate TAD into two tasks: classification and localization using two separate branches. This paper reveals the temporal misalignment between the two tasks hindering further progress. To address this, we propose a new method that gives insights into moment and region perspectives simultaneously to align the two tasks by acquiring reliable proposal quality. For the moment perspective, Boundary Evaluate Module (BEM) is designed which focuses on local appearance and motion evolvement to estimate boundary quality and adopts a multi-scale manner to deal with varied action durations. For the region perspective, we introduce Region Evaluate Module (REM) which uses a new and efficient sampling method for proposal feature representation containing more contextual information compared with point feature to refine category score and proposal boundary. The proposed **B**oundary Evaluate Module and **R**egion **E**valuate **M**odule (BREM) are generic, and they can be easily integrated with other anchor-free TAD methods to achieve superior performance. In our experiments, BREM is combined with two different frameworks and improves the performance on THUMOS14 by 3.6% and 1.0% respectively, reaching a new state-of-the-art (63.6% average *m*AP). Meanwhile, a competitive result of 36.2% average *m*AP is achieved on ActivityNet-1.3 with the consistent improvement of BREM. The codes are released at https://github.com/Junshan233/BREM.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**.

KEYWORDS

Temporal action detection, video analysis, deep neural network

ACM Reference Format:
Junshan Hu, Chaoxu Guo, Liansheng Zhuang, Biao Wang, Tiezheng Ge, Yuning Jiang, and Houqiang Li. 2022. Estimation of Reliable Proposal Quality for Temporal Action Detection. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3503161.3548029

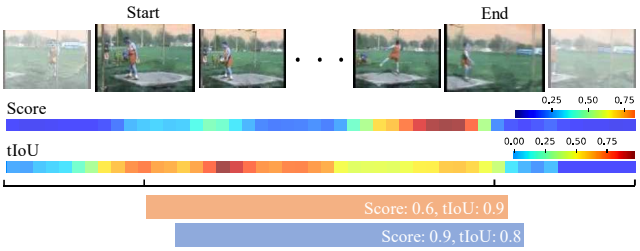

Figure 1: Illustration of misaligned temporal distribution between classification score (*Score*) and localization quality (*tIoU*). Orange and blue blocks indicate two proposals.

1 INTRODUCTION

With the conversion of mainstream media information from text and images into videos, the number of videos on the Internet grows rapidly in recent years. Therefore video analysis evolves into a more important task and attracts much attention from both academy and industry. As a vital area in video analysis, temporal action detection (TAD) aims to localize and recognize action instances in untrimmed long videos. TAD plays an important role in a large number of practical applications, such as video caption [11, 30] and content-based video retrieval [3, 8].

Recently, a number of methods have been proposed to push forward the state-of-the-art of TAD, which can be mainly divided

*Work done during an internship at Alibaba Inc.
†Corresponding Author

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548029

Table 1: Oracle experiment results. G_{score} means proposals scores are replaced by tIoU between proposal and corresponding ground-truth.

G_{score}	0.3	0.4	0.5	0.6	0.7	Avg.
×	60.4	54.9	46.4	35.2	21.5	43.7
✓	93.4	92.0	88.3	82.3	72.8	85.8

into three types: anchor-based [16, 18, 31], bottom-up [14, 15, 37], and anchor-free [12, 26, 35] methods. Although anchor-free methods show stronger competitiveness than others with simple architectures and superior results, they still suffer from the temporal misalignment between the classification and localization tasks.

Current anchor-free frameworks mainly formulate TAD into two tasks: localization and classification. The localization task is designed to generate action proposals, and the classification task is expected to predict action category probabilities which is naturally used as ranking scores in non-maximum suppression (NMS). However, classification and localization tasks usually adopt different training targets. The feature that activates the classification confidence may lack information beneficial to localization, which inevitably leads to misalignment between classification and localization. To illustrate this phenomenon, we present a case on THUMOS14 [10] in Fig. 1, where a proposal with the highest classification score fails to locate the ground truth action. This suggests that the classification score can't accurately represent localization quality. Under this circumstance, accurate proposals may have lower confidence scores and be suppressed by less accurate ones when NMS is conducted. To further demonstrate the importance of accurate score, we replace predicted classification score of action proposals with the actual proposal quality score, which is tIoU between proposal and corresponding ground-truth. As shown in Tab. 1, mAP is greatly improved, which suggests that accurate proposals may not be retrieved due to inaccurate scores.

Recent attempts adopt an additional branch to predict tIoU between proposal and the corresponding ground truth [12] or focus on the center of an action instance [35]. Although notable improvement is obtained, there is still a huge gap between the performance of previous methods and ideal performance. We notice that previous methods mainly rely on the region view which only considers global features of proposals and ignore local appearance and motion evolvement, which increases the difficulty of recognizing boundary location accurateness, especially for actions with long duration.

In this paper, we propose a new framework that gives insights into moment and region views simultaneously to align two tasks by estimating reliable proposal quality. First, we propose Boundary Evaluate Module (BEM) to acquire boundary qualities of proposals from a moment view. Specifically, BEM focuses on local appearance and motion evolvement for predicting the boundary quality of each temporal location which reflects the distance between the current location to the location of the action boundary. Then, the quality of the generated proposal is calculated by its boundary qualities. However, the duration of realistic actions can vary from a few seconds to minutes and the localization quality of short actions is more sensitive to the boundary error than long actions. To address this, multi-scale boundary quality is adopted in BEM in a divide-and-conquer way which assigns a suitable scale for each proposal depending on its duration. For the region view, we propose a simple but effective module named Region Evaluate Module (REM), which employed the sampled features in proposals as the proposal feature representation and refines proposals. In particular, REM obtains aligned features by sampling at three locations within the action proposals, which contain more contextual information beneficial to estimate reliable proposal quality and accurate boundary. The proposed **B**oundary Evaluate Module and **R**egion **E**valuate **M**odule (BREM) are generic, and they can be integrated with other anchor-free TAD methods to achieve better results. To validate the effectiveness of BREM, we conduct experiments on two popular mainstream datasets THUMOS14 [10] and ActivityNet-1.3 [9]. By combining BREM with a basic anchor-free TAD framework proposed by [33], we achieve an absolute improvement of 3.6% *m*AP@Avg on THUMOS14. When integrating with the state-of-the-art TAD framework ActionFormer [35], we achieve a new state-of-the-art (63.6% *m*AP@Avg) on THUMOS14 and competitive result (36.2% *m*AP@Avg) on ActivityNet-1.3.

Overall, the contributions of our paper are following: **1)** Boundary Evaluate Module (BEM) is proposed to predict multi-scale boundary quality and offer proposal quality from a moment perspective. **2)** By introducing Region Evaluate Module (REM), the aligned feature of each proposal are extracted to estimate localization quality in a region view and further refine the locations of action proposals. **3)** The combination of BEM and REM (BREM) makes full use of moment view and region view for estimating reliable proposal quality and it can be easily integrated with other TAD methods with consistent improvement, where a new state-of-the-art result on THUMOS14 and a competitive result on ActivityNet-1.3 are achieved.

2 RELATED WORK

Anchor-Based Method. Anchor-based methods rely on predefined multiple anchors with different durations and the predictions refined from anchors are used as the final results. Inheriting spirits of Faster R-CNN [22], R-C3D [31] first extracts features at each temporal location, then generates proposals and applies proposal-wise pooling, after that it predicts category scores and relative offsets for each anchor. In order to accommodate varied action durations and enrich temporal context, TAL-Net [6] adopts dilation convolution and scale-expanded RoI pooling. GTAN [18] learns a set of Gaussian kernels to model the temporal structure and a weighted pooling is used to extract features. PBRNet [16] progressively refines anchor boundary by three cascaded detection modules: coarse pyramidal detection, refined pyramidal detection, and fine-grained detection. These methods require predefined anchors which are inflexible because of the extreme variation of action duration.

Bottom-up Method. Bottom-up methods predict boundary probability for each temporal location, then combines peak start and end to generate proposals. Such as BSN [15], it predicts start, end, and actionness probabilities and generates proposals, then boundary-sensitive features are constructed to evaluate the confidence of whether a proposal contains an action within its region. BMN [14] employs an end-to-end framework to generate candidates and confidence scores simultaneously. BU-TAL [37] explores the potential

temporal constraints between start, end, and actionness probabilities. Some methods, such as [19, 34, 38] adopt generated proposals by BSN or BMN as inputs and further refine the boundary and predict more accurate category scores. Our method is inspired by bottom-up frameworks, but we utilize boundary probability to estimate proposal quality instead of generating proposals.

Anchor-Free Method. Benefiting from the successful application of the anchor-free object detection [21, 27], anchor-free TAD methods have an increasing interest recently which directly localize action instances without predefined anchors. A2Net [33] explores the combination of anchor-based and anchor-free methods. AFSD [12] is the first purely anchor-free method that extracts salient boundary features using a boundary pooling operator to refine action proposals and a contrastive learning strategy is designed to learn better boundary features. Recent efforts aim to use Transformer for TAD. For example, RTD-Net [26] and TadTR [17] formulate the problem as a set prediction similar to DETR [4]. ActionFormer [35] adopts a minimalist design and replaces convolution networks in the basic anchor-free framework with Transformer networks. Our method belongs to anchor-free methods and is easily combined with anchor-free frameworks to boost the performance.

3 METHOD

Problem Formulation. An untrimmed video can be depicted as a frame sequence $X = \{x_t\}_{t=1}^{T}$ with T frames. Action annotations in video X consists of N_g action instances $\Psi_X = \{\psi_n, y_n\}_{n=1}^{N_g}$, where $\psi_n = (t_s, t_e)$ are timestamp of start and end of the n-th action instance respectively and y_n is the class label. The goal of temporal action detection is to locate boundaries of actions and recognize categories which cover Ψ_X as precisely as possible.

Overview. For an untrimmed video denoted as $X = \{x_t\}_{t=1}^{T}$, a convolution backbone (*e.g.*, I3D [5], C3D [28].) is used to extract 1D temporal feature $f \in \mathbb{R}^{T/v \times C}$, where T, C, v denote video frame, feature channel and stride. Then, up-sample is used to f for acquiring frame level feature f_F. Multi-scale boundary quality of start and end $\hat{P}_{s/e}$ are predicted by f_F (Sec. 3.2). Parallel, several temporal convolutions are used on f to generate the hierarchical feature pyramid. For each hierarchical feature, a shared detection head is applied to predict coarse proposals and category confidences. After that, the aligned feature is extracted for each coarse proposal to refine boundaries and scores (Sec. 3.3). The boundary quality of each proposal is interpolated on $\hat{P}_{s/e}$ according to the temporal location of boundaries.

3.1 Basic Anchor-free Detector

Following recent object detection methods [27] and TAD methods [12, 33], we build a basic anchor-free detector as our baseline, which contains a backbone, a feature pyramid network, and heads for classification and localization.

We adopt I3D network [5] as the backbone since it achieves high performance in action recognition and is widely used in previous action detection methods [12, 37]. The feature output of backbone is denoted as $f \in \mathbb{R}^{T/v \times C}$. Then, f is used to build hierarchical feature pyramid by applying several temporal convolutions. The hierarchical pyramid features are denoted as $\{f^l \in \mathbb{R}^{T/v_l \times C}\}_{l=1}^{L}$, where l means l-th layer of feature pyramid and v_l is the stride for the l-th layer.

The heads for classification and localization consist of several convolution layers which are shared among each pyramid feature. For details, for l-th pyramid feature, classification head produces category score $\hat{y} \in \mathbb{R}^{T/v_l \times C}$, where C is the number of classes. And localization head predicts distance between current temporal location to action boundaries, denoted as $\{(\hat{\delta}_{s,t}, \hat{\delta}_{e,t})\}_{t=1}^{T/v_l}$. Then action detection results are $\{(c_t, s_t, e_t)\}_{t=1}^{T/v_l}$, where

$$c_t = \arg\max(\hat{y}_t),\ s_t = t - \hat{\delta}_{s,t},\ e_t = t + \hat{\delta}_{e,t}. \tag{1}$$

Following AFSD [12], the quality branch is also adopted in the baseline model which is expected to suppress low quality proposals.

Based on this baseline model, we further propose two modules named Boundary Evaluate Module (BEM) and Region Evaluate Module (REM) to address the issue of misalignment between classification confidence and localization accuracy. Noteworthily, the proposed BEM and REM are generic and easily combined not only with the above baseline framework but also with other anchor-free methods that have a similar pipeline. The details of BEM and REM would be explained in the rest of this section.

3.2 Boundary Evaluate Module

As discussed in Sec. 1, the misalignment between classification confidence and localization accuracy would lead detectors to generate inaccurate detection results. To address this, we propose Boundary Evaluate Module (BEM) to extract features and predict action boundary quality maps from a moment view which is complementary to the region view, thus it can provide more reliable quality scores of proposals.

Single-scale Boundary Quality. Boundary quality maps provide localization quality scores for each temporal location. The quality score is only dependent on the distance from the current location to the location of the action boundary of ground truth.

In practice, we set predefined anchors[1] at each temporal location, denoted as $\{a^t\}_{t=1}^{T}$, where $a^t = [t - r/2, t + r/2]$ denotes the anchor at t-th temporal location and r is the predefined anchor size. For a video with action ground truth $\{(t_{s,n}, t_{e,n})\}_{n=1}^{N_g}$, we define start and end region for n-th instance as $g_s^n = [t_{s,n} - r/2, t_{s,n} + r/2]$ and $g_e^n = [t_{e,n} - r/2, t_{e,n} + r/2]$. The boundary quality maps for start boundary and end boundary $P_s, P_e \in \mathbb{R}^T$ are calculated by

$$\begin{aligned} P_s^t &= \max_{n \in N_g} \text{tIoU}(a^t, g_s^n), \\ P_e^t &= \max_{n \in N_g} \text{tIoU}(a^t, g_e^n), \end{aligned} \tag{2}$$

where tIoU is temporal IoU. The parameter r controls the region size, examples for small and large r are shown in Fig. 3 denoted as *Small scale* and *Large scale* separately. In this way, each score in the quality map indicates the location precision of the start or end boundary. In the inference phase, proposal boundary quality is acquired by interpolation at the corresponding temporal location.

Previous works [16, 36] formulate the prediction of boundary probability as a binary classification task that can't reflect the

[1]The *anchor* in anchor-based TAD methods is used to describe potential action instances, while we use *anchor* to calculate boundary quality. Thus our method still belongs to the anchor-free method.

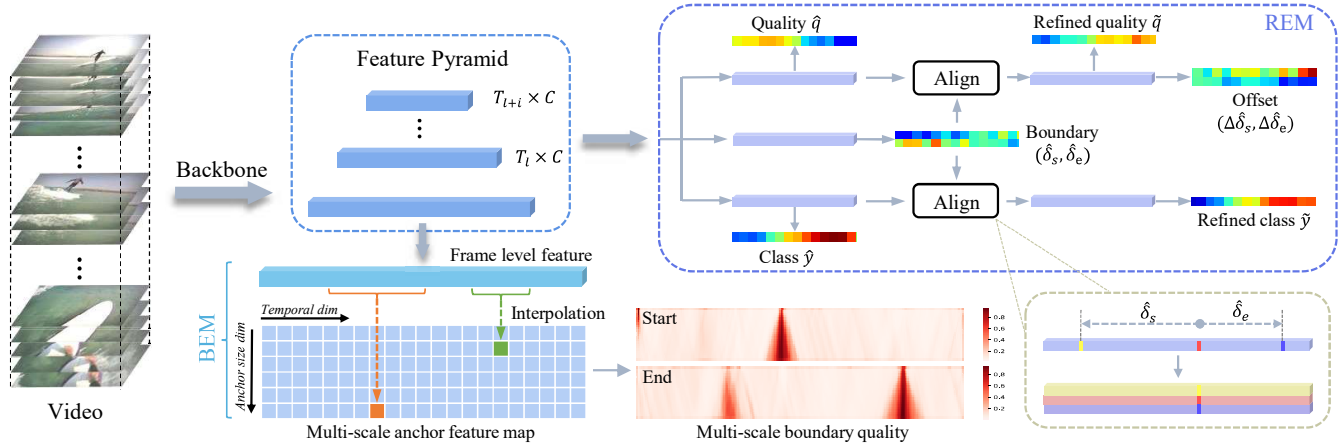

Figure 2: Illustration of the proposed BREM. Untrimmed videos are first fed into the backbone to generate the 1D temporal feature, which is used to construct the feature pyramid and frame-level feature. REM adopts each pyramid feature as input and generates coarse proposals and scores. Then the aligned feature is used for refinement of action location and scores. In parallel, BEM acquires the frame-level feature as input and produces the multi-scale boundary quality map for localization quality prediction.

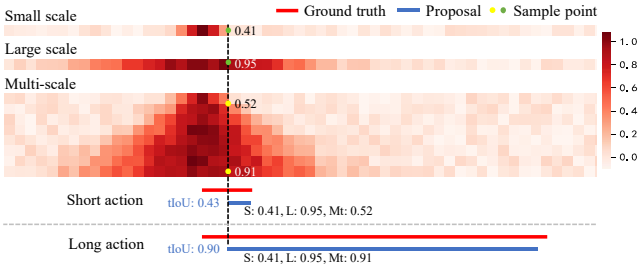

Figure 3: Comparison between single-scale and multi-scale boundary quality maps. For the short proposal and the long proposal, their tIoU are 0.43 **and** 0.90**, and their boundary quality scores of small scale, large scale and multi-scale are** $(0.41, 0.95, 0.52)$ **and** $(0.41, 0.95, 0.91)$.

relative probability differences between two different locations. However, we define precise boundary quality using tIoU between the predefined anchor and boundary region. Moreover, previous works define positive locations by action length (*e.g.*, locations lie in $[s - d/10, s + d/10]$ are positive samples in [16] and [36], where d and s are action length and start location of ground-truth). Thus, the model has to acquire the information of the duration of actions. But it is difficult because of the limited reception field, especially for long actions. So, the definition of boundary quality in Eq. 2 is regardless of the duration of actions. Another weakness of previous works is that they define the action boundary using a small region which leads to that only the proposal boundary closing to the ground-truth boundaries being covered. In this work, we can adjust τ to control the region size. We demonstrate that small region size is harmful to performance in our ablation.

Multi-scale Boundary Quality. Actions with different duration require different sensitivity to the boundary changes. Fig. 3 helps us to illustrate this. If we use *Small scale*, a short proposal and a long proposal (blue lines) with the same localization error of start boundary acquire the same boundary qualities of 0.41, but the actual tIoU of the long proposal is 0.9. Similarly, if we use *Large scale*, these two proposals acquire boundary qualities of 0.95, but the actual tIoU of the short proposal is 0.57. Thus, single-scale boundary quality is suboptimal for varied action duration. The scale should dynamically adapt the duration of actions. To address this, we expand the single-scale boundary quality maps into quality maps with multi-scale anchors. Thus, for a proposal, we can choose a suitable anchor depending on its duration (as yellow points show in Fig. 3).

In detail, start and end boundary quality maps are extended to two dimensions corresponding to temporal time steps and anchor scales, denoting as $P_s, P_e \in \mathbb{R}^{T \times I}$, where I is the number of predefined anchors. We predefine multiple anchors with different size at each temporal location, denoting as $\{A^t\}_{t=1}^{T}$, where $A^t = \{a^{t,1}, \cdots, a^{t,I}\}$ denoting I predefined anchors. The anchor size is defined as

$$R = \{r_{min}, r_{max}, I\}, \tag{3}$$

representing I evenly spaced number from r_{min} to r_{max}, where r_{max} and r_{min} indicate the maximum and minimum anchor scale. In this paper, r_{min} is set as 1 that corresponds to the interval time between adjacent input video frames and r_{max} depends on the distribution of duration of the actions in datasets. We conduct ablation studies about the selection of r_{max} in Sec. 4.2. Thus the i-th anchor at t is $a^{t,i} = [t - r_i/2, t + r_i/2]$. As for a ground-truth $(t_{s,n}, t_{e,n})$, its start and end region for i-th anchor can be denoted as $g_s^{n,i} = [t_{s,n} - r_i/2, t_{s,n} + r_i/2]$ and $g_e^{n,i} = [t_{e,n} - r_i/2, t_{e,n} + r_i/2]$. Then the multi-scale quality maps $P_s, P_e \in \mathbb{R}^{T \times I}$ are calculated by

$$\begin{aligned} P_s^{t,i} &= \max_{n \in N_g} \text{tIoU}(a^{t,i}, g_s^{n,i}) \\ P_e^{t,i} &= \max_{n \in N_g} \text{tIoU}(a^{t,i}, g_e^{n,i}) \end{aligned} \tag{4}$$

In the inference phase, the boundary quality of the proposal is obtained by bilinear interpolation according to the boundaries location and the proposal duration (See Sec.3.4).

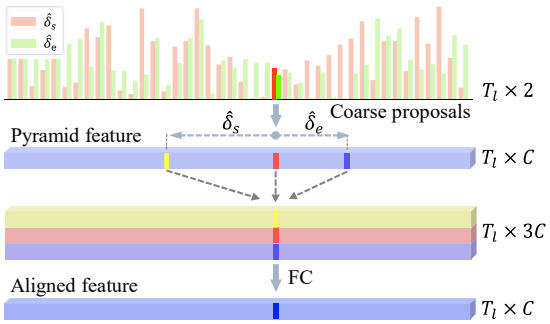

Figure 4: Illustration of feature alignment. According to coarse proposals, sample three features at $\{t - \hat{\delta}_s, t, t + \hat{\delta}_e\}$ and aggregate them by a fully-connected layer.

Implementation. To predict multi-scale boundary quality maps, as shown in Fig.5, the backbone feature is first fed into an upsampling layer and several convolution layers to get the frame-level feature $f_F \in \mathbb{R}^{T\times C}$ with a higher temporal resolution, which is beneficial to predict quality score of the small anchor. Because the anchor scales may have a large range and different scales need different receptive fields, we adopt a parameter-free and efficient method to generate features. In detail, we use linear interpolation in each anchor to obtain the multi-scale anchor feature map, denoted as $M \in \mathbb{R}^{T\times I\times N\times C}$. In particular, for $M(t,i) \in \mathbb{R}^{N\times C}$, we uniformly sample N features in the scope $[t - r_i/2, t + r_i/2]$ from f_F which ensures that the receptive field matches the anchor size. This procedure of interpolation can be efficiently achieved by matrix product[14]. After the multi-scale anchor feature map M is obtained, we apply max pooling on the N sampled features and a 1×1 convolution to extract anchor region representation :

$$M_B = \text{Conv}(\text{MaxPool}(M)), \tag{5}$$

where $M_B \in \mathbb{R}^{T\times I\times C}$. Finally, two boundary score maps are obtained based on M as follows:

$$\begin{aligned} \hat{P}_s &= \sigma(f_s(M_B)) \\ \hat{P}_e &= \sigma(f_e(M_B)) \end{aligned} \tag{6}$$

where $f_s(\cdot)$ and $f_e(\cdot)$ are convolution layers and $\sigma(\cdot)$ is sigmoid function.

Training. We denote label maps for $\hat{P}_s$ and $\hat{P}_e$ as $O_s, O_e \in \mathbb{R}^{T\times I}$ respectively. The label maps is computed by Eq. 4. We take points where $O_{s/e} > 0$ as positive. L2 loss function is adopted to optimize BEM, which is formulated as follows:

$$\begin{aligned} \ell_{bem} &= 0.5 \cdot (\ell_s + \ell_e), \\ \ell_{s/e} &= \frac{1}{|\mathcal{N}^{s/e}|} \sum_{(t,i)\in\mathcal{N}^{s/e}} \left(O^{s/e}_{t,i} - \hat{P}^{s/e}_{t,i}\right)^2, \end{aligned} \tag{7}$$

where $\mathcal{N}^{s/e}$ is the set of positive points.

3.3 Region Evaluate Model

BEM estimates the localization quality of proposals in the moment view that focuses more on local appearance and motion evolvement. Although it achieves considerable improvement, as illustrated in Tab. 4, we believe that feature of the region view can provide rich context information which is beneficial to the prediction of localization quality. Therefore, we propose Region Evaluate Module (REM), as shown in the right part of Fig. 5, which first predicts coarse action proposals and then extracts features of proposals to predict localization quality scores, action categories, and boundary offsets.

Specifically, REM predicts coarse action offset $(\hat{\delta}_s, \hat{\delta}_e)$, action categories $\hat{y}$ and quality score $\hat{q}$ for each temporal location (omitting subscript standing for temporal location for simplicity). For a location t with coarse offset prediction which indicates the distance to start and end of the action boundaries, the corresponding proposal can be denoted as $\hat{\psi} = (t - \hat{\delta}_s, t + \hat{\delta}_e)$. Then three features are sampled from pyramid feature at $\{t - \hat{\delta}_s, t, t + \hat{\delta}_e\}$ via linear interpolation and aggregated by a fully-connected layer. This procedure is illustrated in Fig. 4. Based on the aggregated feature, BEM produces refined boundary offsets $(\Delta\hat{\delta}_s, \Delta\hat{\delta}_e)$, quality scores $\tilde{q}$ and category scores $\tilde{y}$. The final outputs can be obtained by

$$\begin{aligned} y &= 0.5 \cdot (\hat{y} + \tilde{y}), \\ q &= 0.5 \cdot (\hat{q} + \tilde{q}), \\ \psi &= (t - \hat{\delta}_s - 0.5 \cdot \Delta\hat{\delta}_s\hat{w}, t + \hat{\delta}_e + 0.5 \cdot \Delta\hat{\delta}_e\hat{w}) \end{aligned} \tag{8}$$

where ψ, y, q are final action proposal, action category score and location quality score respectively and $\hat{w} = \hat{\delta}_s + \hat{\delta}_e$.

Training. The loss of REM is formulated as:

$$\ell_{rem} = \hat{\ell}_{loc} + \lambda\hat{\ell}_{cls} + \gamma\hat{\ell}_q + \tilde{\ell}_{loc} + \lambda\tilde{\ell}_{cls} + \gamma\tilde{\ell}_q, \tag{9}$$

where λ, γ are loss weight. $\hat{\ell}_{cls}$ and $\tilde{\ell}_{cls}$ are focal loss [13] for category prediction. $\hat{\ell}_q$ and $\tilde{\ell}_q$ are loss of quality prediction, which is implemented by binary cross entropy loss. tIoU between proposal and corresponding ground-truth is adopted as target of quality prediction:

$$\hat{\ell}_q = \frac{1}{|\mathcal{N}_{pos}|} \sum_{t\in\mathcal{N}_{pos}} \text{BCE}(\hat{q}_t, \text{tIoU}(\psi_t, \hat{\psi}_t)), \tag{10}$$

where ψ_t is ground-truth for location t. $\hat{\ell}_{loc}$ is generalized IoU loss [23] for location prediction of initial proposal and $\tilde{\ell}_{loc}$ is L1 loss for offset prediction of the refining stage:

$$\begin{aligned} \hat{\ell}_{loc} &= \frac{1}{|\mathcal{N}_{pos}|} \sum_{t\in\mathcal{N}_{pos}} (1 - \text{GIoU}(\psi_t, \hat{\psi}_t)), \\ \tilde{\ell}_{loc} &= \frac{1}{|\mathcal{N}_{pos}|} \sum_{t\in\mathcal{N}_{pos}} (|\Delta\hat{\delta} - \Delta\delta|) \end{aligned} \tag{11}$$

where $\mathcal{N}_{pos}$ indicates the ground-truth action locations, and $\Delta\delta = 2 \cdot (\delta - \hat{\delta})/\hat{w}$, $\hat{w}$ is coarse proposal length.

3.4 Training and Inference

Training details. Since there are mainly two different strategies for video feature extraction, including online feature extraction [12, 31] and offline feature extraction [14, 15, 19], we adopt different training methods for them. For frameworks using the online feature extractor, BEM and REM are trained jointly with the feature extractor in an end-to-end way. The total train loss function is

$$\ell = \ell_{rem} + \eta\ell_{bem}, \tag{12}$$

where η is used to balance loss. As for methods with the offline feature extractor, since BEM is independent of other branches, we

Table 2: Comparison with state-of-the-art methods on THUMOS14. Average mAP is computed with tIoU thresholds in [0.3 : 0.1 : 0.7]. The best results are in bold. We integrate BREM with two typical frameworks, baseline (*Base*) (Sec. 3.1) and ActionFormer [35]. Our method achieves a new state-of-the-art performance on THUMOS14.

Type	Model	Feature	0.3	0.4	0.5	0.6	0.7	Avg.
Anchor-based	R-C3D [31]	C3D [28]	44.8	35.6	28.9	-	-	-
	GTAN [18]	P3D [20]	57.8	47.2	38.8	-	-	-
	PBRNet [16]	I3D [5]	58.5	54.6	51.3	41.8	29.5	47.1
	A2Net [33]	I3D [5]	58.6	54.1	45.5	32.5	17.2	41.6
	VSGN [36]	TS [24]	66.7	60.4	52.4	41.0	30.4	50.2
	G-TAD [32]	TS [24]	54.5	47.6	40.2	30.8	23.4	39.3
Bottom-up	BSN [15]	TS [24]	53.5	45.0	36.9	28.4	20.0	36.8
	BMN [14]	TS [24]	56.0	47.4	38.8	29.7	20.5	38.5
	BC-GNN [2]	TS [24]	57.1	49.1	40.4	31.2	23.1	40.2
	BU-TAL [37]	I3D [5]	53.9	50.7	45.4	38.0	28.5	43.3
	ContextLoc [38]	I3D [5]	68.3	63.8	54.3	41.8	26.2	50.9
	TCANet [19]	TS [24]	60.6	53.2	44.6	36.8	26.7	44.4
Anchor-free	AFSD [12]	I3D [5]	67.3	62.4	55.5	43.7	31.1	52.0
	RTD-Net [26]	I3D [5]	68.3	62.3	51.9	38.8	23.7	49.0
	TadTR [17]	I3D [5]	62.4	57.4	49.2	37.8	26.3	46.6
	ActionFormer [35]	I3D [5]	75.5	72.5	65.6	56.6	42.7	62.6
	Base	I3D [5]	68.5	63.7	56.6	45.8	31.0	53.1
	Base+BREM	I3D [5]	70.7	66.1	60.0	50.1	36.4	56.7
	ActionFormer+BREM	I3D [5]	**76.5**	**73.2**	**66.9**	**57.7**	**43.7**	**63.6**

individually train BEM and other branches, then combine them in the inference phase for better performance.

Inference. The final outputs of REM is calculated by Eq. 8. Thus, the generated proposals can be denoted as $\{(y, q, \psi)_n\}_{n=1}^{N}$, where $\psi = (t_s, t_e)$ and N is the number of proposals. In order to obtain boundary quality, we define a function that generates index of appropriate anchor scale in multi-scale boundary quality map according to the action duration, denoted as $f(d)$. We adopt a simple linear mapping:

$$\begin{aligned} f(d) &= \frac{r - r_i}{r_{i+1} - r_i} + i, \\ r &= d/\tau, \\ s.t.\ \ r_i &\leqslant r \leqslant r_{i+1}, \end{aligned} \tag{13}$$

where τ is a predefined mapping coefficient. For a proposal, τ controls the anchor size used by it. We explore the influence of τ in our ablation. Then start and end boundary quality are acquired by bilinear interpolation,

$$p_{s,d} = Intep(P_s, (t_s, f(d))),\ p_{e,d} = Intep(P_e, (t_e, f(d))), \tag{14}$$

where $Intep$ is bilinear interpolation and $d = t_e - t_s$ is the length of proposal. After fusing these scores, the final proposals is denoted as $\{(y \cdot q \cdot \sqrt{p_{s,d} \cdot p_{e,d}}, \psi)\}_{n=1}^{N}$.

4 EXPERIMENTS

Dataset. The experiments are conducted on two popularly used datasets, THUMOS14 [10] and ActivityNet-1.3 [9]. THUMOS14 contains 200 untrimmed videos in the validation set and 212 untrimmed videos in the testing set with 20 categories. Following previous works [14, 15, 37], we train our models on the validation set and report the results on the testing set. ActivityNet-1.3 contains 19,994 videos of 200 classes with about 850 video hours. The dataset is split into three subsets, about 50% for training, and 25% for validation and testing. Following [14, 15, 32], the training set is used to train the models, and results are reported on the validate set.

Implementation Details. For THUMOS14 dataset, we sample 10 frames per second (fps) and resize the spatial size to 96×96. Same as the previous works [12, 14], sliding windows are used to generate video clips. Since nearly 98% action instances are less than 25.6 seconds in the dataset, the windows size is set to 256. The sliding windows have a stride of 30 frames in training and 128 frames in testing. The feature extractor is I3D [5] pre-trained in Kinetics. The mean Average Precision (mAP) is used to evaluate performance. The tIoU thresholds of $[0.3 : 0.1 : 0.7]$ are considered for mAP and average mAP. If not noted specifically, we use Adam as optimizer with the weight decay of 10^{-3}. The batch size is set to 8 and the learning rate is 8×10^{-4}. As for loss weight, η, λ, γ are set to 5, 1 and 0.5. The anchor scale R and mapping coefficient τ in BEM are $\{1, 50, 20\}$ and 2. In the testing phase, the outputs of RGB and Flow are averaged. The tIoU threshold of Soft-NMS is set as 0.5.

On ActivityNet-1.3, each video is encoded to 768 frames in temporal length and resized to 96×96 spatial resolution. I3D backbone is pre-trained in Kinetics. mAP with tIoU thresholds $\{0.5, 0.75, 0.95\}$ and average mAP with tIoU thresholds $[0.5 : 0.05 : 0.95]$ are adopted. Optimizer is Adam with weight decay of 10^{-4}. Batch size is 1 and learning rate is 10^{-5} for feature extractor and 10^{-4} for other components. As for loss weight, η, λ, γ are set to 5, 1 and 1 repestively. The anchor scale R and mapping coefficient τ in BEM are $\{1, 130, 22\}$ and 2. The tIoU threshold of Soft-NMS is set to 0.85.

Table 3: Comparison with state-of-the-art methods on ActivityNet-1.3. Average mAP is computed with tIoU thresholds in [0.3 : 0.1 : 0.7]. We integrate BREM with two typical frameworks, baseline (Sec. 3.1) (*Base*) and ActionFormer [35] (*AF*).

Model	Feature	0.5	0.75	0.95	Avg.
Anchor-based					
R-C3D [31]	C3D [28]	26.8	-	-	-
GTAN [18]	P3D [20]	52.6	34.1	8.9	34.3
PBRNet [16]	I3D [5]	54.0	35.0	9.0	35.0
A2Net [33]	I3D [5]	43.6	28.7	3.7	27.8
VSGN [36]	TS [24]	52.4	36.0	8.4	35.1
G-TAD [32]	TS [24]	50.4	34.6	9.0	34.1
Bottom-up					
BSN [15]	TS [24]	46.5	30.0	8.0	30.0
BMN [14]	TS [24]	50.1	34.8	8.3	33.9
BC-GNN [2]	TS [24]	50.6	34.8	**9.4**	34.3
BU-TAL [37]	I3D [5]	43.5	33.9	9.2	30.1
ContextLoc [38]	I3D [5]	**56.0**	35.2	3.6	34.2
TCANet [19]	SlowFast [7]	54.3	**39.1**	8.4	**37.6**
Anchor-free					
AFSD [12]	I3D [5]	52.4	35.3	6.5	34.4
RTD-Net [26]	I3D [5]	47.2	30.7	8.6	30.8
TadTR [17]	I3D [5]	49.1	32.6	8.5	32.3
ActionFormer (TSP)	R(2+1)D [29]	54.1	36.3	7.7	36.0
Base	I3D [5]	52.4	34.3	5.6	33.6
Base+BREM	I3D [5]	52.2	35.4	5.1	34.3
AF+BREM (TSP)	R(2+1)D [29]	53.7	37.9	6.9	36.2

In order to validate the generalizability of our method, we also evaluate the performance when integrating BREM with methods using the offline feature extractor. ActionFormer [35] is the latest anchor-free TAD method that shows strong performance. Thus we integrate BREM with ActionFormer to validate the effectiveness of BREM. The implementation details are shown in our supplement.

4.1 Main Result

In this subsection, we compare our models with state-of-the-art methods, including anchor-based (*e.g.*, R-C3D [31], PBRNet [16], VSGN [36]), bottom-up (*e.g.* BMN [14], TCANet [19]), and anchor-free (*e.g.*, AFSD [12], RTD-Net [26]) methods. And the features used by these methods are also reported for a more fair comparison, including C3D [28], P3D [20], TS [24], I3D [5], and R(2+1)D [29].

The results on the testing set of THUMOS14 are shown in Tab. 2. Our baseline achieves 53.1% *m*AP@Avg outperforming most of the previous methods. Based on the strong baseline, BREM absolutely improves 3.6% from 53.1% to 56.7% on *m*AP@Avg. It can be seen that the proposed BREM acquires improvement on each tIoU threshold compared with the baseline. Especially on high tIoU thresholds, BREM achieves an improvement of 5.4% on *m*AP@0.7. Similarly, integrating BREM with ActionFormer [35] provides a performance gain of 1.3% on *m*AP@0.5 and yields a new state-of-the-art performance of 63.6% on *m*AP@Avg.

Table 4: Effectiveness of BEM and REM. The first row represents the result of the baseline model described in Sec. 3.1.

BEM	REM	0.5	0.6	0.7	Avg.
		47.0	35.4	22.9	44.2
✓		48.9	38.5	27.1	46.4
	✓	47.4	37.4	25.0	45.4
✓	✓	**50.2**	**40.8**	**29.0**	**48.3**

Table 5: The effectiveness of boundary quality. $\{r_{min}, r_{max}, I\}$ represent I evenly spaced numbers from r_{min} to r_{max}. The first row indicates the model without boundary quality.

Type	Anchor size	0.5	0.6	0.7	Avg.
w/o	-	47.0	35.4	22.9	44.2
Single-scale	4	45.2	34.3	21.9	42.6
	16	47.3	37.4	25.9	45.2
	28	**47.8**	**37.5**	**26.4**	**45.4**
	40	47.5	37.4	25.7	45.1
Multi-scale	{1, 10, 20}	47.0	37.1	25.5	45.0
	{1, 20, 20}	47.2	37.6	27.0	45.5
	{1, 40, 20}	48.1	**38.9**	**27.4**	46.3
	{1, 50, 20}	**48.9**	38.5	27.1	**46.4**
	{1, 60, 20}	48.6	38.6	27.2	**46.4**

The results on ActivityNet-1.3 validation set are shown in Tab. 3. Integrating BREM with baseline (*Base*) reaches an average *m*AP of 34.3%, which is 0.7% higher than baseline. And BREM achieves an average *m*AP of 36.2% when combined with ActionFormer (*AF*) using the pre-training method from TSP [1], which is the best result using the features from [29]. It is worthy to note that BREM brings considerable improvement on middle tIoU thresholds, outperforming ActionFormer by 1.6% on *m*AP@0.75. TCANet [19] is the only model better than ours, but it uses the stronger SlowFast feature [7] and refines proposals generated by a strong proposal generation method [14].

4.2 Ablation Study

We conduct ablation experiments on THUMOS14 for the RGB model based on the baseline to validate the effectiveness of our method. The *m*AP at tIoU=0.5, 0.6 and 0.7, and average *m*AP in [0.3 : 0.1 : 0.7] are reported. Each experiment is repeated three times and the average result is presented to obtain more convincing results.

Effectiveness of Model Components. In order to analyze the effectiveness of the proposed BEM and REM, each component is applied in the baseline model gradually. Meanwhile, the result of the combination of BEM and REM is also presented to demonstrate they are complementary to each other. All results are shown in Tab. 4. Obviously, BEM boosts the average *m*AP by 2.2%. The significant improvement brought by BEM confirms that BEM helps to preserve better action proposals based on the more accurate quality score of boundary localization. Meanwhile, REM improves the average *m*AP by 1.2%. This suggests that aligned features are beneficial for refining more accurate boundaries, classification, and quality scores. By combining BEM and REM, the performance is further improved

Table 6: Effectiveness of each component of REM.

Model	0.5	0.6	0.7	Avg.
REM	47.4	**37.4**	**25.0**	**45.4**
w/o offset	47.2	36.7	23.5	44.6
w/o quality	**47.7**	37.1	24.7	45.3
w/o classification	47.0	36.9	24.7	44.9

from 44.2% to 48.3% on *m*AP@Avg. The great complementary result shows that the moment view of BEM and region view of REM are both essential.

Effectiveness of Boundary Quality. In order to demonstrate the effectiveness of boundary quality, we first analyze its importance by introducing single-scale boundary quality. Then the comparison between single-scale and multi-scale boundary quality is conducted to validate the necessity of introducing more anchor scales. Finally, different settings of boundary anchors are explored. Results are shown in Tab. 5. For single-scale boundary quality with anchor size=4, the *m*AP@Avg drops from 44.2% to 42.6%. We conjecture that the reason is that the estimated boundary quality at the most temporal locations can not reflect the actual location quality because of the small anchor size (see Fig. 3 *Small scale*). Increasing the anchor size boosts the performance. The best result is reached with anchor size=28, and further increasing the anchor size harms the performance. For multi-scale boundary quality, we gradually increase the largest anchor size (r_{max}). As shown in Tab. 5, increasing r_{max} improves the performance, and saturation is reached when $r_{max} = 50$ because there are few long actions in the dataset thus too large anchors are rarely used. The above results suggest that our single-scale boundary quality can help preserve better predictions in NMS, but a suitable anchor size has to be carefully chosen. Contrary to single-scale boundary quality, multi-scale boundary quality introduces further improvement by dividing actions into different appropriate anchor scales depending on their duration. It can be seen that the anchor size of $\{1, 50, 20\}$ brings a 1% improvement compared with single-scale boundary quality. Furthermore, it is less sensitive to the choice of anchor size.

Effectiveness of REM. Based on the aligned feature, REM refines the location, category score, and localization quality score of each action proposal. We gradually remove each component to show its effectiveness. The results are shown in Tab. 6. Removing offset, quality, and classification drop the performance by 0.8%, 0.1%, and 0.5% respectively. Refinement of location and category score bring more noticeable improvement to the model than quality score. We preserve quality score refinement in our final model since it can stable the performance and only increases negligible computation. Previous work [12] extracts salient boundary feature by boundary max pooling, while we extract the region feature of the proposal by interpolation which is more efficient and shows competitive performance.

Ablation study on regional feature extraction method in REM. We explore different feature extraction methods in REM, 1) FC: all sampled features in each anchor region are concatenated and a fully connected layer is applied to convert them to the target dimension. 2) Mean: the mean operation is applied to all sampled features. 3) Max: the mean operation in Mean is replaced with max. 4) Mean&Max: Mean feature and Max feature are concatenated and a fully connected layer is applied to convert the dimension of the feature. The results are shown in Tab. 7. FC is commonly used in previous works [14, 25], but reaches the lowest performance in our experiments. Max acquires the best performance of average *m*AP, showing 0.9%, 0.3% and 0.3% advantage against FC, Mean and Mean&Max respectively.

Table 7: Ablation study on regional feature extraction method in REM.

method	0.5	0.6	0.7	Avg.
FC	47.7	37.6	26.3	45.5
Mean	48.2	38.1	**27.4**	46.1
Max	**48.9**	**38.5**	27.1	**46.4**
Mean&Max	48.3	38.1	26.7	46.1

Table 8: Ablation study on mapping coefficient τ in BEM.

τ	0.5	0.6	0.7	Avg.
0.5	48.3	37.3	25.0	45.5
1.0	**49.0**	38.1	26.0	**46.4**
2.0	48.9	**38.5**	**27.1**	**46.4**
4.0	47.1	37.2	25.6	45.0

Ablation study on mapping coefficient τ in BEM. The mapping coefficient τ in BEM controls the corresponding anchor size of the proposal in the inference phase (See Eq. 13). For a proposal, it will use a smaller scale anchor if enlarging τ. We vary the mapping coefficient $\tau \in \{0.5, 1.0, 2.0, 3.0\}$ in the inference phase and report the results in Tab. 8. The performance is stable if τ equals to 1.0 or 2.0. Smaller and larger τ will decrease the performance since the anchor size and the duration of action are not appropriately matched, which also confirms the importance of multi-scale boundary quality.

5 CONCLUSION

In this paper, we reveal the issue of misalignment between localization accuracy and classification score of current TAD methods. To address this, we propose **B**oundary Evaluate Module and **R**egion **E**valuate **M**odule (BREM), which is generic and plug-and-play. In particular, BEM estimates the more reliable proposal quality score by predicting multi-scale boundary quality in a moment perspective. Meanwhile, REM samples region features in action proposals to further refine the action location and quality score in a region perspective. Extensive experiments are conducted on two challenging datasets. Benefiting from the great complementarity of moment and region perspective, BREM achieves state-of-the-art results on THUMOS14 and competitive results on ActivityNet-1.3.

ACKNOWLEDGMENTS

This work was supported in part by Next Generation AI Project of China No.2018AAA0100602, in part to Dr. Liansheng Zhuang by National Natural Science Foundation of China (NSFC) under contract No.U20B2070 and No.61976199, and in part to Dr. Houqiang Li by NSFC under contract No.61836011.

REFERENCES

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. 2021. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3173–3183.

[2] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. 2020. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*. Springer, 121–137.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1728–1738.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

[5] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 4724–4733.

[6] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE Computer Society, 1130–1139.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 6201–6210.

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*. Springer, 214–229.

[9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 961–970.

[10] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes.

[11] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 706–715.

[12] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. 2021. Learning Salient Boundary Feature for Anchor-free Temporal Action Localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 3320–3329.

[13] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2999–3007.

[14] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 3888–3897.

[15] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[16] Qinying Liu and Zilei Wang. 2020. Progressive Boundary Refinement Network for Temporal Action Detection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 11612–11619.

[17] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Song Bai, and Xiang Bai. 2021. End-to-end temporal action detection with transformer. *arXiv preprint arXiv:2106.10271* (2021).

[18] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. 2019. Gaussian Temporal Awareness Networks for Action Localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 344–353.

[19] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. 2021. Temporal Context Aggregation Network for Temporal Action Proposal Refinement. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 485–494.

[20] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 5534–5542.

[21] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 779–788.

[22] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 91–99.

[23] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. 2019. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 658–666.

[24] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 568–576.

[25] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan. 2021. BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2602–2610.

[26] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. 2021. Relaxed Transformer Decoders for Direct Action Proposal Generation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 13506–13515.

[27] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 9626–9635.

[28] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 4489–4497.

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

[30] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018. Bidirectional Attentive Fusion With Context Gating for Dense Video Captioning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE Computer Society, 7190–7198.

[31] Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 5794–5803.

[32] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali K. Thabet, and Bernard Ghanem. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 10153–10162.

[33] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. 2020. Revisiting Anchor Mechanisms for Temporal Action Localization. *IEEE Trans. Image Process.* 29 (2020), 8535–8548.

[34] Runhao Zeng, Wenbing Huang, Chuang Gan, Mingkui Tan, Yu Rong, Peilin Zhao, and Junzhou Huang. 2019. Graph Convolutional Networks for Temporal Action Localization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 7093–7102. https://doi.org/10.1109/ICCV.2019.00719

[35] Chenlin Zhang, Jianxin Wu, and Yin Li. 2022. ActionFormer: Localizing Moments of Actions with Transformers.

[36] Chen Zhao, Ali K Thabet, and Bernard Ghanem. 2021. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13658–13667.

[37] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Bottom-up temporal action localization with mutual regularization. In *European Conference on Computer Vision*. Springer, 539–555.

[38] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. 2021. Enriching Local and Global Contexts for Temporal Action Localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13516–13525.

A EXPERIMENTAL DETAIL

The proposed **B**oundary Evaluate Module and **R**egion **E**valuate **M**odule (BREM) is generic, and it is easily combined with other anchor-free frameworks to achieve better results. Fig. 5 illustrates the overall architecture of anchor free methods combined with BREM, where BEM is integrated as an extra component and REM is adopted to replace the original detection head. In our experiments, BREM is combined with two typical frameworks, a basic anchor-free framework (denoted as *Base*) and ActionFormer [35]. The implementation details are described in this section.

A.1 The architecture of Boundary Evaluate Module

The implementation of Boundary Evaluate Module (BEM) is shown in Tab 9. The input feature of BEM is frame level feature $f_F \in \mathbb{R}^{C\times T}$ with the time resolution same as the input of backbone, which preserves more detail information of appearance and motion. In the experiments, the number of sample points N is set to 14 for the balance of efficiency and effectiveness.

A.2 The architecture of Region Evaluate Module

As Figure 2 shown in our main paper, the inputs of REM are feature pyramid denoted as $\{f^l \in \mathbb{R}^{T/o_l\times C}\}_{l=1}^{L}$. For simplicity, we omit the superscript standing for pyramid layers in the following. For a feature of pyramid (denoted as f), BREM predicts coarse action offset $\hat{\delta}$, action categories $\hat{y}$ and quality score $\hat{q}$ by the equations

$$\begin{aligned}\hat{\delta} &= h_o(g_o(f)),\\ \hat{y} &= \sigma(h_c(g_c(f))),\\ \hat{q} &= \sigma(h_q(g_o(f))),\end{aligned} \tag{15}$$

where $g_o(\cdot), g_c(\cdot)$ are hidden convolution layer, and $h_o(\cdot), h_c(\cdot), h_q(\cdot)$ are convolution layer with output channels of 2 (coarse offset to start and end of action), C (the number of categories) and 1 respectively. The hidden layer of $\hat{q}$ is shared with offset prediction branch which is usually adopted in previous work [1]. As for refined boundary offsets $\Delta\hat{\delta}$, quality scores $\tilde{q}$ and category scores $\tilde{y}$, they use a similar method as the above description except that the input feature is the aligned feature by "Align" module (see Figure 4 in the main paper). This is formulated as

$$\begin{aligned}f' &= \text{Align}(f),\ \Delta\hat{\delta} = h'_o(g'_o(f')),\\ \tilde{y} &= \sigma(h'_c(g'_c(f'))),\ \tilde{q} = \sigma(h'_q(g'_o(f'))).\end{aligned} \tag{16}$$

Finally, the refined action prediction is obtained via Eq. 8 in the main paper.

A.3 Combination with *Base*

Detailed architecture. We adopt a previous successful feature pyramid proposed by AFSD [12] and its architecture is shown in Fig. 6. I3D [5] is used to extract the semantic feature of videos and the feature of $4th$ and $5th$ stage (C4 and C5) are used to generate a feature pyramid. Meanwhile, an up-sample layer and a convolutional layer are used to produce the frame-level feature. Finally, pyramid features and frame level feature are fed into REM and BEM, respectively.

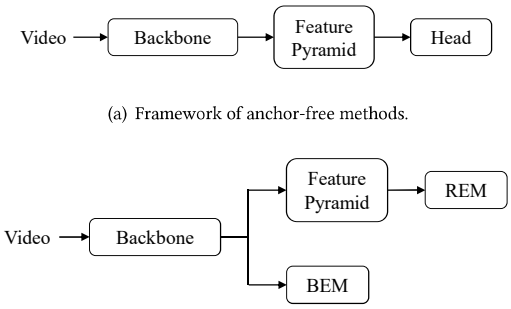

Figure 5: The framework of anchor-free methods and BREM.

Table 9: The implementation of Boundary Evaluate Module. T, I, and N are length of input feature, number of anchors, and number of sample points. *Intep* denotes sampling features within each anchor.

layer	kernel	output dim	act	output size
$conv1d_1$	3	256	*relu*	$256 \times T$
Intep				$256 \times N \times T \times I$
MaxPool				$256 \times 1 \times T \times I$
Squeeze				$256 \times T \times I$
$conv2d_1$	1	128	*relu*	$128 \times T \times I$
$conv2d_2$	3	128	*relu*	$128 \times T \times I$
$conv2d_3$	1	2	*sigmoid*	$2 \times T \times I$

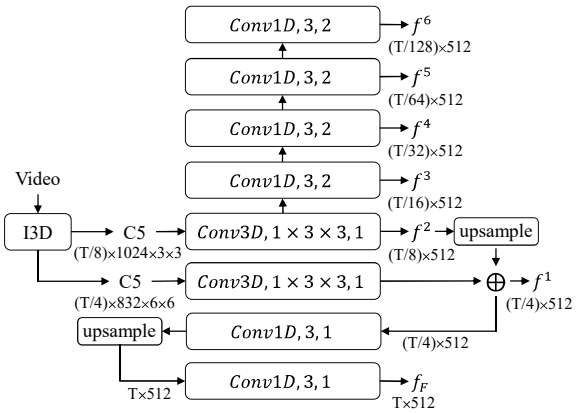

Figure 6: The architecture of feature encoder of *Base*. The convolutional layer is denoted as *Conv*, *kernel size*, *stride*, and the channel of all convolutional layers is 512. $\oplus$ denotes element-wise summation.

A.4 Combination with ActionFormer

Detailed architecture. Contrary to *Base*, ActionFormer [35] uses off-the-shelf features. Frame level feature f_F is generated by applying a *Conv1D* with kernel size=3 and stride=1 on pre-encoded video features. Then f_F is fed into BEM and the original head of ActionFormer is replaced with REM.

Table 10: Results of different mapping functions.

Mapping	0.5	0.6	0.7	Avg.
w/o BEM	47.0	35.4	22.9	44.2
$f(d\|\tau = 2)$	**48.9**	**38.5**	**27.1**	**46.4**
$g(d\|D = 6)$	46.0	36.0	23.7	44.0
$g(d\|D = 18)$	47.4	37.2	25.3	45.4
$g(d\|D = 31)$	48.3	**37.8**	**25.9**	**45.8**
$g(d\|D = 43)$	**48.4**	37.5	25.3	45.6

Implementation. Since the video feature f is pre-encoded, REM and BEN are separately trained for better performance and convergence. Other training details are same as ActionFormer [35].

B ADDITIONAL EXPERIMENT RESULTS

B.1 Additional Ablation Study of Multi-scale Boundary Quality

In this section, we conduct additional ablation studies to explore the effectiveness of multi-scale boundary quality compared with single-scale boundary quality.

The proposed method uses a linear mapping function $f(d)$ to generate the index of appropriate anchor scale in the multi-scale boundary quality map according to the action duration. In this section, we explore a special mapping function, denoted as $g(d)$,

$$\begin{aligned} g(d) &= \frac{r - r_i}{r_{i+1} - r_i} + i, \\ r &= D, \\ s.t.\ \ r_i \leqslant r \leqslant r_{i+1}, & r_{min} \leqslant D \leqslant r_{max}, \end{aligned} \tag{17}$$

where D is a hyper-parameter which indicates the anchor size used in the inference phase. This mean that all proposals use the same anchor size D. Unlike $f(d)$ that assigns proposals to anchors with appropriate size, $g(d)$ assigns all proposals to a same anchor. The comparison between $f(d)$ and $g(d)$ can demonstrate the effectiveness of assigning proposals of different duration to appropriate anchors. In the experiments on THUMOS14, we set 20 anchors from 1 to 50 with even interval, denoted as $R = \{r_{min}, r_{max}, I\} = \{1, 50, 20\}$. We replace $f(d)$ with $g(d)$ and vary $D \in \{6, 18, 31, 43\}$, and the results are shown in Tab. 10. Using $g(d)$, the best result is reached when $D = 31$, which is lower than using $f(d|\tau = 2)$ (-0.6% in average mAP). The results confirm that dealing with proposals with different duration by using anchors of different sizes is effective to acquire reliable proposal quality.

B.2 Integrating BREM with More Methods

In order to demonstrate the effectiveness of the proposed method, BREM is integrated with more TAD methods. The results are shown in Tab. 11 and Tab. 12. According to these results, BREM achieves consistent improvement regardless of TAD methods and the improvement is more significant when combining with weakly methods, e.g. A2Net.

Table 11: Comparison of state-of-the-art methods with and without BREM on THUMOS14. * indicates ours implementation.

Method	0.3	0.4	0.5	0.6	0.7	Avg.
A2Net*	56.5	51.1	43.0	31.1	16.6	39.7
A2Net+BREM	62.0	56.9	47.0	34.2	21.1	44.2
Base	68.5	63.7	56.6	45.8	31.0	53.1
Base+BREM	70.7	66.1	60.0	50.1	36.4	56.7
AF	75.5	72.5	65.6	56.6	42.7	62.6
AF+BREM	76.5	73.2	66.9	57.7	43.7	63.6

Table 12: Comparison of state-of-the-art methods with and without BREM on ActivityNet. * indicates ours implementation.

Method	0.5	0.75	0.95	Avg.
A2Net*	43.1	28.6	4.9	28.0
A2Net+BREM	46.0	31.0	5.4	30.2
Baseline	52.4	34.3	5.6	33.6
Baseline+BREM	52.2	35.4	5.1	34.3
AF*	53.6	35.9	7.3	35.2
AF+BREM	52.8	36.6	6.9	35.5
AF (TSP)	54.1	36.3	7.7	36.0
AF+BREM (TSP)	53.7	37.9	6.9	36.2
TCANet*	51.7	36.3	10.3	35.5
TCANet+BREM	52.2	36.3	10.3	35.7

Table 13: Comparison of inference speed between our method and other methods.

Method	Speed (ms)	Memory (MB)
AFSD	63.1	1495
Baseline	45.2	1215
Baseline+BREM	55.6	1533
ActionFormer	2109.9	1971
ActionFormer+BREM	2180.4	2251

B.3 Comparison on Inference Speed

We provide a comparison of inference speed of different methods with and without BREM on THUMOS14. All results are tested on a video with 25.6s, 30fps and on a server with an NVIDIA Tesla V100 GPU. As Tab. 13 shown, Baseline+BREM acquires 12% relative speed improvement compared to AFSD, which is previous state-of-the-art method. And the additional memory usage of BREM is negligible because almost 80% of the memory is consumed by the backbone. As for ActionFormer, because of time-consuming feature extraction (98% of total time), the inference speed is lower than Baseline and BREM increases only 3.3% inference time. Above results show that BREM can bring considerable improvement with negligible memory and runtime cost.