Information Processing and Management 60 (2023) 103141

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Anchor-free temporal action localization via Progressive Boundary-aware Boosting

Yepeng Tang [a,b,1], Weining Wang [c], Yanwu Yang [d], Chunjie Zhang [a,b,*], Jing Liu [c,e]

[a] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing, 100044, China
[b] Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China
[c] National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[d] School of Management, Huazhong University of Science and Technology, Wuhan, 430074, China
[e] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Keywords:
Temporal action localization
Anchor-free
Video understanding

ABSTRACT

Enormous untrimmed videos from the real world are difficult to analyze and manage. Temporal action localization algorithms can help us to locate and recognize human activity clips in untrimmed videos. Recently, anchor-free temporal action localization methods have gained increasing attention due to small computational costs and no complex hyperparameters of pre-set anchors. Although the performance has been significantly improved, most existing anchor-free temporal action localization methods still suffer from inaccurate action boundary predictions. In this paper, we want to alleviate the above problem through boundary refinement and temporal context aggregation. To this end, a novel Progressive Boundary-aware Boosting Network (PBBNet) is proposed for anchor-free temporal action localization. The PBBNet consists of three main modules: Temporal Context-aware Module (TCM), Instance-wise Boundary-aware Module (IBM), and Frame-wise Progressive Boundary-aware Module (FPBM). The TCM aggregates the temporal context information and provides features for the IBM and the FPBM. The IBM generates multi-scale video features to predict action results coarsely. Compared with IBM, the FPBM focuses on instance features corresponding to action predictions and uses more supervision information for boundary regression. Given action results from IBM, the FPBM uses a progressive boosting strategy to refine the boundary predictions multiple times with supervision from weak to strong. Extensive experiments on three benchmark datasets THUMOS14, ActivityNet-v1.3 and HACS show our PPBNet outperforms all existing anchor-free methods. Further, our PPBNet achieves state-of-the-art performance (72.5% mAP at tIoU = 0.5) on THUMOS14 dataset.

1. Introduction

As the cost of photography decreases, information is often stored in the form of video data in many scenarios (Rani & Kumar, 2020; Zhao, Zhang, et al., 2021; Hassani, Ershadi, & Mohebi, 2022). Unlike text, image, and audio, video data with both spatial and temporal information is more complex, especially long-duration untrimmed videos in real world. To analyze the untrimmed videos, we often focus on a theme of interest, such as human activities, animal activities, or object movements. Then, computers can trim and recognize video snippets about the assigned theme from the untrimmed videos with deep learning algorithms. It can

[*] Corresponding author.
E-mail address: cjzhang@bjtu.edu.cn (C. Zhang).

[1] This work was done when Yepeng Tang was an intern at the Institute of Automation, Chinese Academy of Sciences.

https://doi.org/10.1016/j.ipm.2022.103141
Received 31 July 2022; Received in revised form 23 October 2022; Accepted 24 October 2022
Available online 8 November 2022

help us to analyze and process the huge amount of real-world video data efficiently. Among the subjects, there is no doubt that human is the most essential and important theme. Furthermore, videos containing human behaviors appear most commonly thus collected easily. Hence, most studies on untrimmed video understanding focus on human actions. Temporal action localization (TAL), a vital and fundamental video understanding task, is developed to study on how to locate and classify every video clip that may contain an human action instance in the untrimmed videos. The TAL algorithms can be applied to a variety of scenarios (Zhao, Torralba, Torresani, & Yan, 2019; Hosono, Sawada, Sun, Hayase, & Shimamura, 2020; Dave et al., 2022; Li et al., 2022; Alkanat, Akdag, Bondarev, & de With, 2022), including surveillance, business recommendations, autonomous driving, etc. The TAL task has attracted increasing attention in recent years.

Most existing TAL methods rely on predefined or dense temporal anchors, which lead to a large number of redundant action proposals and high computational complexity. Given an untrimmed video with N frames, existing actionness-based methods based on dense anchors (Lin, Liu, Li, Ding, & Wen, 2019; Lin et al., 2020; Wang, Zhang, Zheng, & Pan, 2022) usually perform an exhaustive enumeration of all boundary combinations and evaluate their confidence scores by building a $T \times T$ two-dimensional proposal map, which generates $\frac{T\times(T-1)}{2}$ action proposals. The anchor-based methods (Xu, Das, & Saenko, 2017; Chao et al., 2018; Gao et al., 2020) pre-set anchors with N different scales for each temporal location and regress action boundaries on the top of these anchors, which produce $N \times T$ action proposals. If using multi-scale features, the number of action proposals will be greater. Besides, the performances of anchor-based methods are sensitive to the hyperparameters of anchors. Complex hyperparameter adjustments limit the application of anchor-based TAL methods.

For efficiency, anchor-free TAL methods take each temporal location as a target point to regress the distances between the location and two action boundaries, which only generate T action proposals and do not preset anchors. Compared with the above TAL methods, anchor-free methods require fewer hyperparameters and lower computing costs. It enables the anchor-free approach to be a competitive alternative. Existing anchor-free TAL methods have made preliminary explorations on how to design an anchor-free TAL framework. Lin et al. (2021) proposed a basic coarse-to-fine anchor-free TAL method named AFSD. The AFSD method design a boundary max-pooling strategy to build a boundary salient refinement module that can refine the coarse boundary predictions. Some anchor-free methods (Zhao et al., 2022; Zhang, Wu, & Li, 2022) achieve good performance on the benchmark datasets with the help of self-attention mechanisms.

However, most of the existing anchor-free TAL models still suffer from imprecise action boundary predictions. (1) Because of generating only a small number of proposals, the anchor-free TAL methods have natural inferiority in action boundary prediction. The AFSD adopts a coarse-to-fine framework that effectively improves the precision of boundary prediction, but its refinement strategy is relatively simple and does not fully utilize the boundary neighborhood information. It leads to ambiguous boundary prediction when the boundary context is complex. The AFSD model only refines the action proposals once. It also limits the improvement of boundary prediction. (2) The anchor-free TAL method regresses action boundaries directly upon temporal location, which relies on the ability of models to capture temporal context information. The capability of existing anchor-free TAL models for temporal modeling is still unsatisfactory.

To solve the above problems, we propose a novel anchor-free temporal action localization framework named Progressive Boundary-aware Boosting Network (PBBNet) to detect action instances through boundary refinement and temporal context aggregation. With the powerful capability of capturing temporal context, the PPBNet can generate high-quality boundary predictions through progressive boundary-aware boosting. To be specific, the PBBNet consists of three main modules: Temporal Context-aware Module (TCM), Instance-wise Boundary-aware Module (IBM), and Frame-wise Progressive Boundary-aware Module (FPBM). The TCM is used to aggregate the temporal context information. It generates coarse-grained and fine-grained aggregated features for the IBM and the FPBM, respectively. The IBM is used to locate the approximate temporal position of action instances. It generates multi-scale features by a pyramid network and predicts action boundary and category on each location of multi-scale features. The FPBM is used to refine the coarse boundary predictions from IBM. Compared with IBM, the FPBM focuses on action instance features and uses more supervision information. Besides, the FPBM extracts instance features according to the action predictions and adopts a progressive boundary-aware boosting strategy, where action boundaries are regressed multiple times in the frame level with supervision from weak to strong.

The main contributions of this paper can be summarized as follows:

- We propose a novel Progressive Boundary-aware Boosting Network (PBBNet) for anchor-free temporal action localization. Unlike existing coarse-to-fine anchor-free methods, the refinement of PBBNet directly focuses on action instance features and adopts a progressive boundary-aware boosting strategy to refine boundary predictions with supervision from weak to strong.
- A novel temporal context-aware module is designed to improve the capability of capturing temporal context information. It first transforms 1D temporal features into 2D space to capture local information by 2D convolution and then uses self-attention to aggregate global information in 1D space.
- Comprehensive experiments are performed on THUMOS14, ActivityNet-v1.3 and HACS datasets. Our proposed PBBNet achieves state-of-the-art performance on THUMOS14 dataset. Moreover, our PBBNet outperforms all existing anchor-free TAL models on ActivityNet-v1.3 and HACS datasets.

The remaining sections of the paper are organized as follows. In Section 2, we introduce related works about anchor-free temporal action localization. In Section 3, we describe the details of our PBBNet method. In Section 4, experiments and analysis on three benchmark datasets are provided. Finally, Section 5 makes the conclusions of this paper.

2. Related work

Due to the explosion of video data, it is expected that computers can help us process, analyze and utilize this rich yet redundant video data through artificial intelligence technologies. Thus, video understanding algorithms gained a lot of attention and have been developed significantly. Akin to image classification (Zhang, Jiang, et al., 2017; Dosovitskiy et al., 2020), video classification (i.e., action recognition) algorithms (Feichtenhofer, Fan, Malik, & He, 2019) have achieved high accuracy. However, video classification methods can only handle short-duration trimmed videos and are not capable of handling long-duration untrimmed videos. It limits real-world applications of video understanding technologies. To solve this issue, the temporal action localization (TAL) task focuses on processing untrimmed videos. It can trim and recognize video snippets about human activities from untrimmed videos with the help of deep learning. Most TAL methods (Liu, Ma, Zhang, Liu, & Chang, 2019; Chen et al., 2019; Zhao et al., 2020; Tan, Tang, Wang, & Wu, 2021) separate this task into two subtasks: temporal action proposal generation and video classification. They first generate temporal action proposals (like bounding boxes in object detection) with high confidence scores. Then, they classify these action proposals using advanced video classification algorithms (Zhao, Zhang, et al., 2017). Some TAL methods (Lin, Zhao, & Shou, 2017; Long et al., 2019; Lin et al., 2021) are one-stage algorithms, which locate and classify action instances from untrimmed videos simultaneously. However, the performances of one-stage TAL algorithms are often lower than two-stage TAL algorithms. In fact, both the one-stage and two-stage approaches need to locate action instances by generating action proposals. In this paper, we focus on how to generate action proposals. The TAL methods can be divided into three categories: actionness-based methods, anchor-based methods and anchor-free methods. We will discuss these methods separately in the following.

Actionness-based TAL methods. Actionness-based TAL methods adopt a bottom-up method to predict action proposals. The 'actionness' means the probability that each video clip contains an action instance in an untrimmed video. Besides, the probability that each frame is a boundary frame or action frame is often used. For example, BSN (Lin, Zhao, Su, Wang, & Yang, 2018) predicts the starting boundary, ending boundary and action probabilities of each temporal location and chooses the temporal points with high boundary probabilities. It can generate many action proposals by combining the starting points and the ending points. With the help of action probability, BSN (Lin et al., 2018) extracts proposal features to predict their actionness scores and finally obtains flexible action proposals. However, it is inefficient due to choosing and combining boundary points. To solve this problem, BMN (Lin et al., 2019) presets dense temporal anchors, i.e., use action proposals by combining each pair of different temporal points. It produces a 2D proposal feature map and predicts the actionness scores of each proposal in the 2D map. Besides, it also predicts the probabilities of starting boundary and ending boundary for each location like BSN (Lin et al., 2018). Finally, it combines the actionness scores and the corresponding boundary probabilities to obtain high-quality proposals. Afterward, many actionness-based TAL methods (Lin et al., 2020; Su, Gan, Wu, Qiao, & Yan, 2021; Qing et al., 2021; Yang et al., 2022; Wang et al., 2022) follow it and use dense anchors to predict action instances by different actionness evaluation strategies. For example, RCL (Wang et al., 2022) adopts a fully continuous and scale-invariant sampling strategy to generate dense action anchors, and recurrently predicts the action instances in the untrimmed videos. Although these actionness-based TAL methods can achieve high performances, the cost of computing a large number of dense anchors is very high.

Anchor-based TAL methods. Anchor-based TAL methods use a top-down way to predict action proposals. These methods preset a certain number of temporal anchors for each temporal location and regress action boundaries based on these pre-defined anchors. Inspired by the object detection method (Ren, He, Girshick, & Sun, 2015), R-C3D (Xu et al., 2017) use pre-defined anchors to generate proposal features by 3D fully convolution operations. Then it produces the results by feature pooling and classifying. TAL (Chao et al., 2018) focuses on the receptive fields of anchors with different scales and improves the performance through the multi-tower framework and multi-stream fusion feature. To solve the robustness problem caused by pre-defined anchor scales, GTAN (Long et al., 2019) uses the Gaussian kernels to help its model generate action proposals with different duration flexibly. For capturing global temporal context information, RapNet (Gao et al., 2020) uses the relation-aware attention blocks to produce a series of multi-scale temporal pyramid features and generate anchor-based proposals for each location. It can achieve high performance through boundary adjustment and proposal ranking. Although these anchor-based TAL methods can generate good action predictions, complex hyperparameters setting of pre-defined anchors limits their applications. Due to being freedom of pre-defined anchors, our method is more efficient than the anchor-based TAL methods.

Anchor-free TAL methods. Anchor-free TAL methods detect action instances by regressing the distances between the location and two action boundaries for each temporal location. These methods only generate a small number of action proposals without pre-defined anchors, which need fewer hyperparameters and lower computational costs. AFSD (Lin et al., 2021) designs a boundary max-pooling strategy and uses an end-to-end method to train the TAL model. Compared with many actionness-based and anchor-based methods, it obtains comparable performances. TRA (Zhao et al., 2022) focuses on the utilization of temporal relations in anchor-free methods. To capture temporal-aware information, it designs three temporal modeling modules including a temporal self-attention module, a multiple temporal aggregation module and a graph relation module. Both AFSD and TRA adopt a coarse-to-fine framework to detect action instances in untrimmed videos. Unlike AFSD and TRA, Actionformer (Zhang et al., 2022) uses an efficient transformer framework to detect action instances and directly produces the final prediction results without a refinement stage. With the help of the transformer framework, it achieves exciting performances on the main datasets. Although existing anchor-free TAL methods achieve high performance, they still suffer from inaccurate action boundary predictions.

To solve this problem, we propose a novel anchor-free TAL model named progressive boundary-aware boosting network. Compared with AFSD (Lin et al., 2021) and TRA (Zhao et al., 2022), our method focuses on instance features instead of entire video features when refining the action predictions. Besides, our model adopts a progressive boundary-aware boosting strategy to refine the results step by step, while AFSD and TRA only give one chance to refine the predictions. Inspired by Actionformer (Zhang

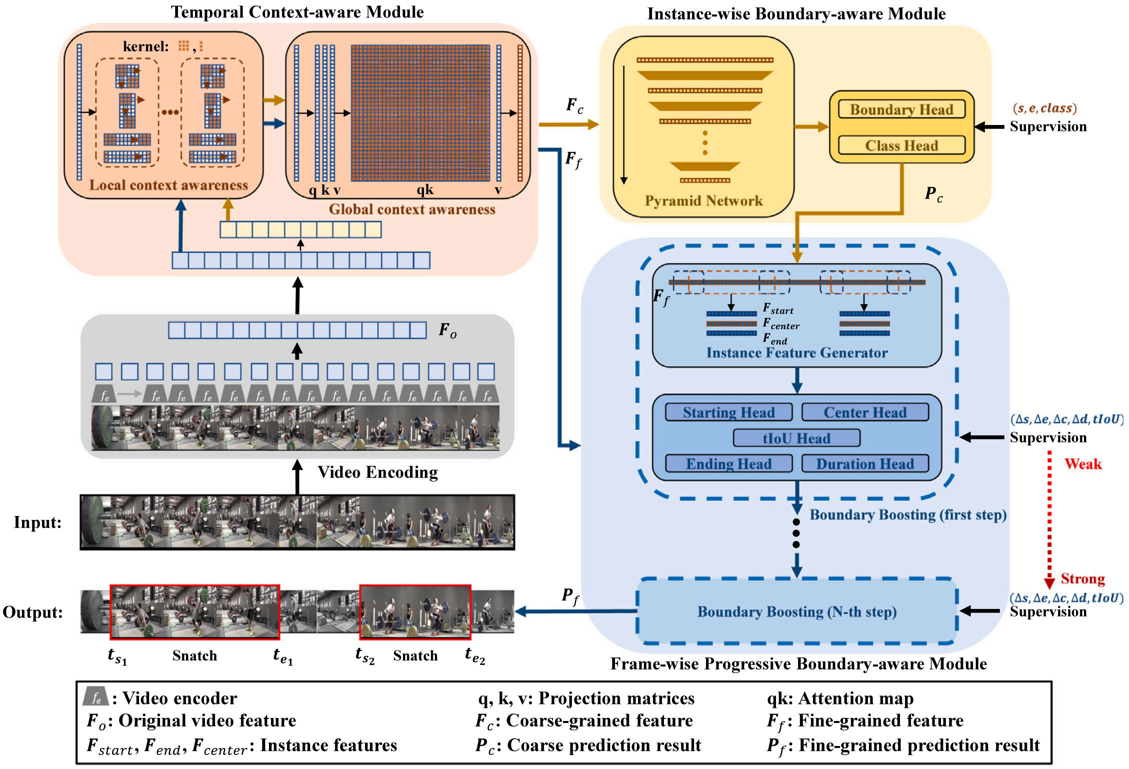

Temporal Context-aware Module
kernel:
Local context awareness
q k v
qk
v
Global context awareness
Instance-wise Boundary-aware Module
Pyramid Network
Boundary Head
Class Head
$(s, e, class)$
Supervision
F_c
F_f
P_c
F_o
Video Encoding
Input:
Output:
t_{s_1}
Snatch
t_{e_1}
t_{s_2}
Snatch
t_{e_2}
P_f
F_f
F_{start}
F_{center}
F_{end}
Instance Feature Generator
Starting Head
Center Head
tIoU Head
Ending Head
Duration Head
$(\Delta s, \Delta e, \Delta c, \Delta d, tIoU)$
Supervision
Weak
Boundary Boosting (first step)
Strong
Boundary Boosting (N-th step)
$(\Delta s, \Delta e, \Delta c, \Delta d, tIoU)$
Supervision
Frame-wise Progressive Boundary-aware Module
: Video encoder
F_o: Original video feature
$F_{start}, F_{end}, F_{center}$: Instance features
q, k, v: Projection matrices
F_c: Coarse-grained feature
P_c: Coarse prediction result
qk: Attention map
F_f: Fine-grained feature
P_f: Fine-grained prediction result

Fig. 1. An overview of our proposed PBBNet. The PBBNet consists of the video encoder, the temporal context-aware module, the instance-wise boundary-aware module and the frame-wise progressive boundary-aware module. The video encoder is used to extract the video feature from an untrimmed video. The temporal context-aware module is used to aggregate the temporal context information. The instance-wise boundary-aware module generates multi-scale features and predicts action instances coarsely. The frame-wise progressive boundary-aware module generates instance features corresponding to the predictions and uses N-step progressive boundary-aware boosting to refine the boundary predictions with supervision from weak to strong. Finally, we can obtain high-quality action instance predictions. The coarse-grained and fine-grained predicting flows are yellow and blue. The demo video has two "snatch" action instances, which can be detected by our method accurately.

et al., 2022), we use a transformer framework to build our model. With a useful refinement module, our method can obtain better predictions than Actionformer. Moreover, we design a valid temporal context-aware module to fully use temporal context information. Combined with the above, our model has a strong ability to capture temporal context information and can generate high-quality boundary predictions through the progressive boosting strategy. Hence, our method can achieve better performance compared with the existing anchor-free TAL methods.

3. Method

In this section, our proposed Progressive Boundary-aware Boosting Network (PPBNet) is introduced in detail. As shown in Fig. 1, our PBBNet consists of the video encoder, the temporal context-aware module, the instance-wise boundary-aware module and the frame-wise progressive boundary-aware module. Given an untrimmed video, we first extract the original feature by a video encoder. We use the temporal context-aware module to capture temporal context information. It outputs coarse-grained and fine-grained feature sequences for the instance-wise boundary-aware module and frame-wise progressive boundary-aware module, respectively. The instance-wise boundary-aware module uses a pyramid network to process the coarse-grained feature sequences and generates action predictions by a classification head and a boundary regression head. The frame-wise progressive boundary-aware module boosts the action boundary predictions step by step with supervision from weak to strong. In each step, it extracts the instance features corresponding to the action predictions and uses multiple regression heads to refine the predictions in the frame level. Finally, we obtain high-quality predictions of action instances.

3.1. Video encoding

Denote an untrimmed video as $\boldsymbol{V} = \{I_i\}_{i=1}^{N_f}$. N_f is the number of video frames. The temporal boundaries and classes of action instances are labeled. For video encoding, we first set a frame sampling interval N_σ to spilt the untrimmed video into a series of video snippets $\{s_i\}_{i=1}^{N_s}$, where $s_i = \{I_i\}_{i=1}^{N_\sigma}$ and $N_s = N_f / N_\sigma$. Following previous methods (Zhang, Li, Zhao, Zhang, & Yan, 2016;

Lin et al., 2021; Qing et al., 2021; Zhang et al., 2022; Wang et al., 2022), we use the state-of-the-art action recognition models (such as I3D (Carreira & Zisserman, 2017) and SlowFast (Feichtenhofer et al., 2019)) to extract feature $\boldsymbol{F}_{snippet} \in \mathbb{R}^{1\times C_o}$ from each video snippet. C_o is the number of feature channels. We concatenate these consecutive snippet features as original feature sequence $\boldsymbol{F}_o \in \mathbb{R}^{N_s\times C_o}$. Following previous methods (Alwassel, Giancola, & Ghanem, 2021; Liu, Bai, & Bai, 2022), we use different video encoders for different untrimmed videos datasets. More details will be given in the experiment section.

3.2. Temporal context-aware module

As each snippet feature is encoded independently, it only considers the temporal context information in the video snippet level. The temporal relationship between video clips remains to be explored. In other words, we need to capture the temporal context information of the entire untrimmed video. Inspired by the advanced previous methods (Zhang, Li, et al., 2017; Xie, Girshick, Dollár, Tu, & He, 2017; Liu, Hu, et al., 2021; Qing et al., 2021; Zhang et al., 2022), we utilize the combination of convolution and self-attention operations to construct the temporal context-aware module.

Denote $\boldsymbol{F} \in \mathbb{R}^{T\times C}$ as the input feature sequence. We use a temporal sliding window to sample a series of consecutive feature sequences $\{\boldsymbol{F}_w\}_{i=1}^{N_w}$ without overlap, where $\boldsymbol{F}_w \in \mathbb{R}^{L_W\times C}$, L_w is the length of sliding window and $T = L_W \times N_w$. Concatenating these feature sequences in chronological order, a higher dimensional feature $\boldsymbol{F}' \in \mathbb{R}^{N_w\times L_W\times C}$ is generated. Setting different sliding windows to extract the high-dimensional features, we use a group of 2D convolution layers with different kernel sizes to generate a elementary temporal-correlated feature. It can be written by

$$\boldsymbol{F}_{local} = f_{flatten}(f^1_{conv2d}(\boldsymbol{F}'_1, [K_{h_1}, K_{w_1}]) + f^2_{conv2d}(\boldsymbol{F}'_2, [K_{h_2}, K_{w_2}]) + \cdots + f^m_{conv2d}(\boldsymbol{F}'_m, [K_{h_3}, K_{w_3}])), \tag{1}$$

where $\boldsymbol{F}'_i$ is generated by a length L_{W_i} sliding window and $[K_{h_i}, K_{w_i}]$ is the kernel size of 2D convolution layer f^i_{conv2d} $(i = 1, 2, \dots, m)$. With the flatten operation $f_{flatten}$, the shape of generated feature $\boldsymbol{F}_{local}$ is $\mathbb{R}^{T\times C'}$ and C' is the number of output channels in the last 2D convolution layer. To improve efficiency, we use the 2D deformable convolution operation (Dai et al., 2017) to perform this instead of both sliding window and 2D convolution operations. The offset sizes of deformable convolution are set according to the length of sliding windows and the kernel sizes. Simultaneously, multiple groups of 2D deformable convolution operations are adopted to improve the performance. Finally, we use two convolution kernels with kernel sizes 1 × 3 and 3 × 3. Three sliding windows are adopted with lengths 3, 6 and 9. With the set of sliding window lengths $\{L_{W_i}\}_{i=1}^4 = \{3, 3, 6, 9\}$, a 2D deformable convolution operation can be written by $f_{flatten}(f^1_{conv2d}(\boldsymbol{F}'_1, [1, 3]) + f^2_{conv2d}(\boldsymbol{F}'_2, [3, 3]) + f^3_{conv2d}(\boldsymbol{F}'_3, [3, 3]) + f^4_{conv2d}(\boldsymbol{F}'_4, [3, 3]))$. We stack 3 groups of these 2D deformable convolution operations, where the output channel dimensions are set to 384, 512 and 512, respectively.

By the above mentioned, we capture the correlations between adjacent and non-adjacent snippet features under different sizes of receptive fields, and aggregate the local temporal contextual information. We further use the self-attention mechanism (Vaswani et al., 2017; Wang, Girshick, Gupta, & He, 2018) to model the global temporal context information. The advanced temporal-correlated feature is generated by the following formula:

$$\boldsymbol{F}_{global} = f_{softmax}(\frac{\boldsymbol{F}_{local}\boldsymbol{M}_Q(\boldsymbol{F}_{local}\boldsymbol{M}_K)^T}{\sqrt[4]{C_Q \times C_K}})\boldsymbol{F}_{local}\boldsymbol{M}_V, \tag{2}$$

where $\boldsymbol{F}_{global} \in \mathbb{R}^{T\times C'}$ and $f_{softmax}$ is the softmax operation. $\boldsymbol{M}_Q \in \mathbb{R}^{C'\times C_Q}$, $\boldsymbol{M}_K \in \mathbb{R}^{C'\times C_K}$ and $\boldsymbol{M}_V \in \mathbb{R}^{C'\times C_V}$ are the projection matrices and $C_Q = C_K$. The feature $\boldsymbol{F}_{global}$ is taken as output.

Specially, for the original feature sequence $\boldsymbol{F}_o$ from Section 3.1, we first re-scale its temporal length to coarse-grained size T_c and fine-grained scale T_f, where $T_c < T_f$. Aggregating the local and global temporal context information, we finally generate a coarse-grained feature $\boldsymbol{F}_c \in \mathbb{R}^{T_c\times C_c}$ and a fine-grained feature $\boldsymbol{F}_f \in \mathbb{R}^{T_f\times C_f}$ for the instance-wise boundary-aware module and frame-wise progressive boundary-aware module, respectively. For reducing computing costs, we use 1D temporal convolution operation instead of the operations in formula (1) when generating the coarse-grained feature. This controls the number of anchor points in the coarse-grained prediction and ensures sufficient temporal resolution in the fine-grained prediction. Besides, it also reduces the amount of calculation.

3.3. Instance-wise boundary-aware module

The instance-wise boundary-aware module is used to generate coarse prediction of temporal action instances. Given a video feature sequence, we utilize the pyramid network to generate multi-scale feature sequences and detect the action instances by boundary and class heads.

Pyramid Network We use N_L depthwise 1D convolution layers as downsampling operators to construct the pyramid networks. Following advanced methods (Wu et al., 2021; Touvron, Cord, Sablayrolles, Synnaeve, & Jégou, 2021; Zhang et al., 2022), we add a local self-attention layer (Choromanski et al., 2020) before each convolution layer. As shown in Algorithm 1, we input the feature $\boldsymbol{F}_c \in \mathbb{R}^{T_c\times C_c}$ and obtain a set of the multi-scale feature sequences $\{\boldsymbol{F}^{(0)}, \boldsymbol{F}^{(1)}, \dots, \boldsymbol{F}^{(N_L)}\}$, where $\boldsymbol{F}^{(i)} \in \mathbb{R}^{T^{(i)}\times C}, (i = 0, 1, \dots, N_L)$.

Boundary Head The boundary head is built by the 1D convolution layers. To achieve better performance, multiple 2D convolution groups from Formula (1) are used. For each feature sequences from the pyramid network, we detect each location j on the temporal dimension and predict the distances (d_s^j, d_e^j) from each location to the starting and ending boundary of an action instance. For example, given the feature sequence $\boldsymbol{F}^{(i)} \in \mathbb{R}^{T^{(i)}\times C}$, $T^{(i)}$ pairs of starting and ending boundary prediction results are generated and can be written by

$$\boldsymbol{R}^{(i)}_{boundary} = \{\boldsymbol{P}^j_{boundary} = (t_j - d_s^j, t_j + d_e^j)\}_{j=1}^{T^{(i)}}. \tag{3}$$

Algorithm 1: Pyramid Network to generate multi-scale video features.

```
Input:
    Feature sequence $\boldsymbol{F}_{input} \in \mathbb{R}^{T \times C}$.
Output:
    Multi-scale feature sequences $\{\boldsymbol{F}^{(0)}, \boldsymbol{F}^{(1)}, \cdots, \boldsymbol{F}^{(N_L)}\}$ and $\boldsymbol{F}^{(i)} \in \mathbb{R}^{\frac{T}{2^i} \times C} (i = 0, 1, \cdots, N_L)$.
Set $\mathcal{S} = \{\}$ and $i = 1$;
Let $\boldsymbol{F}^{(0)} = \boldsymbol{F}_{input}$ and add $\boldsymbol{F}^{(0)}$ into $\mathcal{S}$;
Let learnable factors $\alpha_1$ and $\alpha_2$ be initialized to 0;
while $i <= N_L$ do
    $\boldsymbol{F}_m = f_{LN}(\boldsymbol{F}^{(i-1)})$;   # $f_{LN}$ is the LayerNorm operation.
    $\boldsymbol{F}_m = \alpha_1 f_{SA}(\boldsymbol{F}_m) + \boldsymbol{F}_m$;   # $f_{SA}$ is the Self-Attention operation and $\alpha_1$ is learnable factor.
    $\boldsymbol{F}_m = f_{LN}(\boldsymbol{F}_m)$;
    $\boldsymbol{F}_m = \alpha_2 f_{MLP}(\boldsymbol{F}_m) + \boldsymbol{F}_m$;   # $f_{MLP}$ is the combination of fully-connected layers and GELU operations.
    $\boldsymbol{F}^{(i)} = f_{conv1d}(\boldsymbol{F}_m)$;   # $f_{conv1d}$ is the depthwise 1D convolution operation for downsampling.
    Add $\boldsymbol{F}^{(i)}$ into $\mathcal{S}$ and $i = i + 1$;
end while
return Pyramid feature set $\mathcal{S}$.
```

Class Head The main architecture of class head is similar to the boundary head. For each feature sequences from the pyramid network, we also examine each location j on the temporal dimension. The difference is that the class head predicts the probability of N_c action classes. For the feature sequence $\boldsymbol{F}^{(i)} \in \mathbb{R}^{T^{(i)} \times C}$, $T^{(i)}$ probability sequences are generated and can be written by

$$\boldsymbol{R}^{(i)}_{class} = \{\boldsymbol{P}^{j}_{class} = (p^j_1, p^j_2, \dots, p^j_{N_c})\}^{T^{(i)}}_{j=1}. \tag{4}$$

Supervision Given an untrimmed video, all action instances that it contains are annotated with the boundary locations and class id. For every temporal location j of the multi-scale feature sequences, the instance-wise boundary-aware module outputs the boundary prediction $\boldsymbol{P}^{j}_{boundary}$ and category prediction $\boldsymbol{P}^{j}_{class}$. The loss function is given by

$$L_{coarse} = \sum \left(\frac{f_b(j)}{N_t^+} L_{boundary}(\boldsymbol{P}^{j}_{boundary}, \boldsymbol{P}^{G}_{boundary}) + \frac{1}{N_t} L_{class}(\boldsymbol{P}^{j}_{class}, \boldsymbol{P}^{G}_{class}) \right) \tag{5}$$

where $L_{boundary}$ is the advanced IoU loss (Rezatofighi et al., 2019; Zheng et al., 2020) and L_{class} is the focal loss (Lin, Goyal, Girshick, He, & Dollár, 2017). $\boldsymbol{P}^{G}_{boundary}$ and $\boldsymbol{P}^{G}_{class}$ are the groundtruth of action boundaries and classes, respectively. $f_b(j)$ is a two-value function. If the temporal point j is located in an action region, the point j is considered as a positive sample and $f_b(j) = 1$. Otherwise, it is a negative sample and $f_b(j) = 0$. N_t is denoted as the total number of temporal points and N_t^+ is the number of points regarded as positive samples.

3.4. Frame-wise progressive boundary-aware module

The frame-wise progressive boundary-aware module (FPBM) is used to refine the coarse boundary predictions from IBM. Compared with IBM, the FPBM focuses on action instance features instead of video features and uses more supervision information. Inspired by the advanced methods (Nie et al., 2019; Qing et al., 2021; Pan, Li, Zhang, & Tang, 2021; Wang et al., 2022), we adopt a progressive boundary-aware boosting strategy to refine the boundary predictions step by step with supervision from weak to strong. In each boundary-aware boosting step, the boundary-aware boosting block first extracts the instance features corresponding to action predictions and uses multiple regression heads to refine the results through various types of supervision information. Besides, with the fine-grained video feature, the action boundary predictions can be boosted in the frame level. The specific structure of boundary-aware boosting block is described as follows.

Instance Feature Generator For boundary-aware boosting, we construct the instance features by temporal boundary context and action internal context. Three types of instance features including starting boundary feature, ending boundary feature and center feature are generated. Given the fine-grained video feature $\boldsymbol{F}_f \in \mathbb{R}^{T_f \times C_f}$ and boundary predictions $\boldsymbol{P}_{boundary} = (t_s, t_e)$, the duration of predicted action instance is $d = t_e - t_s$. The starting boundary region, ending boundary region and center region are set to $[t_s - \frac{d}{2}, t_s + \frac{d}{2}]$, $[t_e - \frac{d}{2}, t_e + \frac{d}{2}]$ and $[t_s, t_e]$, respectively. We extract the feature sequences corresponding to these temporal regions from $\boldsymbol{F}_f$ and obtain $\boldsymbol{F}_{start} \in \mathbb{R}^{d \times C_f}$, $\boldsymbol{F}_{end} \in \mathbb{R}^{d \times C_f}$ and $\boldsymbol{F}_{center} \in \mathbb{R}^{d \times C_f}$. As the duration of boundary predictions maybe different, all instance features are projected to the same shape by RoI Align (He, Gkioxari, Dollár, & Girshick, 2017).

Multi-Head Block Multiple groups of 1D convolution layers construct the multi-head block. Each group of 1D convolution layers is used to regress a fine-grained offset. We consider the deviations in terms of starting, ending, center position, and duration. Given the instance features, the fine-grained predictions can be written by

$$\begin{cases} \Delta t_s = f_{conv1d}(f_{ReLu}(f_{conv1d}(\boldsymbol{F}_{start}))) \\ \Delta t_e = f_{conv1d}(f_{ReLu}(f_{conv1d}(\boldsymbol{F}_{end}))) \\ \Delta t_c = f_{conv1d}(f_{ReLu}(f_{conv1d}(\boldsymbol{F}_{start} \| \boldsymbol{F}_{center} \| \boldsymbol{F}_{end}))) \\ \Delta d = f_{conv1d}(f_{ReLu}(f_{conv1d}(\boldsymbol{F}_{start} \| \boldsymbol{F}_{center} \| \boldsymbol{F}_{end}))) \end{cases} \tag{6}$$

where $\cdot \parallel \cdot$ means the temporal concatenation operation. Then, we fuse these predictions and obtain the final results. Given the coarse starting boundary t_s and starting boundary offset Δt_s, a new starting boundary t'_{s_1} can be computed by $(t_s - \Delta t_s \cdot d)$. Besides, we can obtain the coarse duration prediction $d = t_e - t_s$ and the coarse center location $\frac{t_s + t_e}{2}$. Given the center offset Δt_c and duration offset Δd, new center location and duration prediction can be written by $\frac{t_s+t_e}{2} - \Delta t_c \cdot d$ and $d \cdot exp(\Delta d)$. We can also obtain a new starting boundary $t'_{s_2} = \frac{t_s+t_e}{2} - \Delta t_c \cdot d - \frac{d \cdot exp(\Delta d)}{2}$. The final starting boundary t'_s is computed by $\frac{1}{2}(t'_{s_1} + t'_{s_2})$. In the same way, we can get the ending boundary prediction t'_e. Finally, the fine-grained prediction can be written by

$$t'_s = \frac{1}{2}(t_s - \Delta t_s \cdot d) + \frac{1}{2}(\frac{t_s + t_e}{2} - \Delta t_c \cdot d - \frac{d \cdot exp(\Delta d)}{2}) = \frac{3t_s + t_e - (2\Delta t_s + 2\Delta t_c + exp(\Delta d)) \cdot d}{4}, \tag{7}$$

$$t'_e = \frac{1}{2}(t_e - \Delta t_e \cdot d) + \frac{1}{2}(\frac{t_s + t_e}{2} - \Delta t_c \cdot d + \frac{d \cdot exp(\Delta d)}{2}) = \frac{t_s + 3t_e - (2\Delta t_s + 2\Delta t_c - exp(\Delta d)) \cdot d}{4}. \tag{8}$$

In previous methods (Lin et al., 2019, 2020; Qing et al., 2021), temporal intersection over union (tiou) between the action instance proposal and the groundtruth is regard as the actionness probability of this predicted action instance. To achieve better performance, we also predict the actionness probability y to re-rank the predicted action instances, where $y = f_{conv1d}(f_{ReLu}(f_{conv1d}(\boldsymbol{F}_{start} \| \boldsymbol{F}_{center} \| \boldsymbol{F}_{end})))$.

Supervision Given the ith predicted action instances (t^i_s, t^i_e) from the coarse predictions and the corresponding groundtruth $(t^{G_i}_s, t^{G_i}_e)$. The supervision information is given by

$$\begin{cases} \Delta t^{G_i}_s = \frac{t^i_s - t^{G_i}_s}{t^i_e - t^i_s}, \quad \Delta t^{G_i}_e = \frac{t^i_e - t^{G_i}_e}{t^i_e - t^i_s}, \quad \Delta t^{G_i}_c = \frac{t^i_c - t^{G_i}_c}{t^i_e - t^i_s}, \quad \Delta d^{G_i} = ln(\frac{t^{G_i}_e - t^{G_i}_s}{t^i_e - t^i_s}) \\ y^{G_i} = \frac{min\{t^i_e, t^{G_i}_e\} - max\{t^i_s, t^{G_i}_s\}}{t^i_e + t^{G_i}_e - t^{G_i}_s - t^i_s - min\{t^i_e, t^{G_i}_e\} - max\{t^i_s, t^{G_i}_s\}} \end{cases} \tag{9}$$

Then, the loss function is designed as

$$L_{refine} = \sum_i \left(\frac{f'_b(i)}{N^+_p} \sum_x L_{reg}(x^i, x^{G_i}) + \frac{1}{N_p} L_{reg}(y^i, y^{G_i}) \right), x \in \{\Delta t_s, \Delta t_e, \Delta t_c, \Delta d\} \tag{10}$$

where L_{reg} is the SmoothL1 loss. $f'_b(i)$ is a two-value function. If the groundtruth y^{G_i} is larger than a positive threshold τ, the ith predicted action instance is considered as a positive sample and $f_b(j) = 1$. Otherwise, $f_b(j) = 0$. N_p is denoted as the number of all predicted action instances and N^+_p is the number of the positive samples. We set the positive threshold τ from low to high, which can progressive boosting the boundary predictions. In ablation studies, we discuss the importance of progressive boosting strategy and explore the impact of positive thresholds.

4. Experiments

In this section, the main results and analysis of our proposed PBBNet are described in detail. We conducted experiments on the publicly recognized datasets, including THUMOS14, ActivityNet-v1.3 and HACS.

4.1. Dataset and evaluation metric

THUMOS14 dataset (Jiang et al., 2014) It provides 413 untrimmed videos, where 20 classes of action instances are labeled. The duration of total videos comes to 30 h and the number of total action instances is 6365. Specially, 200 labeled untrimmed videos are used for training and 213 labeled untrimmed videos are used for testing.

ActivityNet-v1.3 dataset (Caba Heilbron, Escorcia, Ghanem, & Carlos Niebles, 2015) It provides about 20k untrimmed videos, where 200 classes of action instances are labeled. The duration of total videos comes to about 700 h and the number of total action instances is about 30k. Specially, 50% of untrimmed videos are used as training set, 25% of untrimmed videos are used as validation set, and 25% of untrimmed videos are used as testing set. In particular, the annotation information of the testing set is not publicly available.

HACS dataset (Zhao et al., 2019) It provides about 50k untrimmed videos, where 200 classes of action instances are labeled. The duration of total videos comes to about 2k hours and the number of total action instances is about 140k. Specially, about 38k untrimmed videos are used as training set, 6k untrimmed videos are used as validation set, and 6k untrimmed videos are used as testing set. In particular, the annotation information of the testing set is also not publicly available.

Evaluation Metric Following previous works (Lin et al., 2019; Tan et al., 2021; Zhang et al., 2022), we use the mean Average Precision (mAP) under a series of Temporal Intersection over Union (tIoU) thresholds to evaluate the TAL models. For THUMOS14 dataset, tIoU thresholds is set to {0.3, 0.4, 0.5, 0.6, 0.7}. For ActivityNet-v1.3 and HACS datasets, tIoU thresholds are set to {0.5, 0.75, 0.95} and the average mAP is computed under tIoU thresholds from 0.5 to 0.95 with step size of 0.05.

4.2. Implementation details

Following previous methods (Lin et al., 2021; Zhang et al., 2022; Qing et al., 2021), we choose the two-stream I3D network, TSP network and SlowFast network as video encoders for THUMOS14, ActivityNet-v1.3 and HACS datasets, respectively. We choose Adam as the optimizer. For the stabilization of training, we first optimize the network with the loss function in formula (5). For THUMOS14 dataset, batch size, epoch number, learning rate and weight decay are set to 2, 55(including 5 linear warm-up epochs), 10^{-5} (with a cosine decay) and 10^{-4}, respectively. T_c and T_f are set to 2304 and 3500, respectively. For ActivityNet-v1.3 and HACS dataset, we set batch size, epoch number, learning rate and weight decay to 16, 15 (including 5 linear warm-up epochs), 10^{-3} and 10^{-4}, respectively. We set T_c to 768 and let T_f be 1000. Then, we train the network using the loss function in formula (10). For efficiency, we select top-K coarse predictions for training through multi-class non-maximum suppression. As the THUMOS14 dataset has only a small number of untrimmed videos, we add some external dropout operations to avoid overfitting. Following previous methods (Lin et al., 2021; Zhang et al., 2022), we use the external classifier to improve the accuracy of classification on ActivityNet-v1.3 and HACS datasets. For THUMOS14 dataset, batch size, epoch number, learning rate and weight decay are set to 4, 5, 10^{-6} and 10^{-5}, respectively. For ActivityNet-v1.3 and HACS datasets, we set batch size, epoch number, learning rate and weight decay to 16, 9, 10^{-4} and 10^{-5}, respectively.

4.3. Comparison with the state-of-the-art TAL methods

Performance comparison experiments between our PPBNet and other the state-of-the-art TAL methods are performed on THUMOS14, ActivityNet-v1.3 and HACS datasets. There are different video feature encoders utilized in existing TAL methods, including TSN (Simonyan & Zisserman, 2014), C3D (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015), I3D (Carreira & Zisserman, 2017), P3D (Qiu, Yao, & Mei, 2017), Slowfast (Feichtenhofer et al., 2019) and TSP (Alwassel et al., 2021). For clarity, we indicate the video features used for each experimental result.

Performance comparison on THUMOS14 dataset

Table 1 gives the comparison results on THUMOS14 dataset. Our proposed PPBNet achieves **state-of-the-art performance** with 82.6% mAP at tIoU=0.3, 79.1% mAP at tIoU=0.4, 72.5% mAP at tIoU=0.5, 59.5% mAP at tIoU=0.6 and 45.4% mAP at tIoU=0.7. Our method outperforms other SOTA temporal action localization methods, including actionness-based TAL methods (Shou et al., 2017; Zhao, Xiong, et al., 2017; Lin et al., 2018, 2019, 2020; Xu et al., 2020; Bai et al., 2020; Zhao et al., 2020; Sridhar et al., 2021; Tan et al., 2021; Yang et al., 2021; Su et al., 2021; Liu, Hu, et al., 2021; Liu, Wang, et al., 2021; Yang et al., 2022; Wang et al., 2022), anchor-based TAL methods (Xu et al., 2017; Chao et al., 2018; Long et al., 2019; Liu & Wang, 2020), anchor-free methods (Yang et al., 2020; Lin et al., 2021; Zhao et al., 2022; Zhang et al., 2022), and other TAL methods (Shou et al., 2016; Zeng et al., 2019; Zhu et al., 2021; Zhao, Thabet, & Ghanem, 2021; Chen et al., 2022; Liu et al., 2022). Especially, our PBBNet improve 12.5% mAP and 21.2% mAP over the best actionness-based model (Yang et al., 2022) and anchor-based TAL model (Su et al., 2020) at tIoU threshold 0.5, respectively.

In anchor-free TAL methods, both AFSD (Lin et al., 2021) and TRA (Zhao et al., 2022) adopt a coarse-to-fine framework to detect action instances in untrimmed videos. Due to full utilization of temporal information, TRA (Zhao et al., 2022) perform 1.9% mAP at tIoU=0.5 higher than AFSD (Lin et al., 2021). However, they ignore the importance of progressive refinement and only give their models one chance to refine the predictions. Considering this issue, our PBBNet uses a progressive boosting strategy to refine the results step by step. Thus, our method can gain 17% and 15.1% mAP improvement over AFSD (Lin et al., 2021) and TRA (Zhao et al., 2022), respectively. Besides, Actionformer (Zhang et al., 2022), an efficient transformer anchor-free TAL model, has powerful time-series information acquisition capability and obtains impressive performance crossing the 70% mAP at tIoU=0.5 for the first time. But lacking an effective refinement module limits its performance. We focus on both the utilization of temporal context information and boosting the prediction results progressively. Therefore, our PBBNet can still outperform Actionformer (Zhang et al., 2022) by 1.5% mAP at tIoU=0.5.

Performance comparison on ActivityNet-v1.3 dataset In Table 2, our method is compared with actionness-based methods (Shou et al., 2017; Lin et al., 2018, 2019; Xu et al., 2020; Bai et al., 2020; Zhao et al., 2020; Su et al., 2021; Tan et al., 2021; Liu, Hu, et al., 2021; Liu, Wang, et al., 2021; Qing et al., 2021; Sridhar et al., 2021; Yang et al., 2022; Wang et al., 2022), anchor-based methods (Xu et al., 2017; Chao et al., 2018; Long et al., 2019; Liu & Wang, 2020; Su et al., 2020), anchor-free methods (Yang et al., 2020; Lin et al., 2021; Zhao et al., 2022; Zhang et al., 2022), and others (Zeng et al., 2019; Zhu et al., 2021; Zhao, Thabet, & Ghanem, 2021; Chen et al., 2022; Liu et al., 2022). our PBBNet achieve 55.7 mAP at tIoU=0.5, 38.2 mAP at tIoU=0.75, 7.5 mAP at tIoU=0.95, and 37.1% average mAP, which is a good result compared with other SOTA TAL methods.

In anchor-free TAL methods, Actionformer (Zhang et al., 2022) uses I3D features to obtain a high performance (35.6% average mAP) and outperforms the SOTA anchor-free methods (AFSD (Lin et al., 2021) and TRA (Zhao et al., 2022)) by 1.2% average mAP. With the help of the pre-training TAL feature encoder TSP (Alwassel et al., 2021), Actionformer (Zhang et al., 2022) get a better result and achieves 36.0% average mAP. For a fair comparison, we also use TSP feature for our PBBNet. Due to progressive refinement and fully using temporal context information, our method exhibits 1.1% average mAP improvement over Actionformer (Zhang

Table 1
Performance comparison between our proposed PPBNet and other SOTA temporal action localization methods on THUMOS14 dataset in terms of mAP at differnt IoU thresholds.

	Method	Feature	mAP at different tIoUs				
			0.3	0.4	0.5	0.6	0.7
Others	S-CNN (Shou, Wang, & Chang, 2016)	C3D	36.3	28.7	19.0	10.3	5.3
	P-GCN (Zeng et al., 2019)	I3D	63.6	57.8	49.1	–	–
	P-GCN (Zeng et al., 2019)	TSP	69.1	63.3	53.5	40.4	26.0
	ContextLoc (Zhu, Tang, Wang, Zheng, & Hua, 2021)	I3D	68.3	63.8	54.3	41.8	26.2
	VSGN (Zhao, Thabet, & Ghanem, 2021)	TSN	66.7	60.4	52.4	41.0	30.4
	DCAN (Chen, Zheng, Wang, & Lu, 2022)	TSN	68.2	62.7	54.1	43.9	32.6
	E2E-TAD (Liu et al., 2022)	I3D	59.6	54.5	47.0	37.8	26.5
	E2E-TAD (Liu et al., 2022)	Slowfast	69.4	64.3	56.0	46.4	34.9
Actionness-based	CDC (Shou, Chan, Zareian, Miyazawa, & Chang, 2017)	C3D	40.1	29.4	23.3	13.1	7.9
	SSN (Zhao, Xiong, et al., 2017)	TSN	51.0	41.0	29.8	–	–
	BSN (Lin et al., 2018)	TSN	53.5	45.0	36.9	28.4	20.0
	BMN (Lin et al., 2019)	TSN	56.0	47.4	38.8	29.7	20.5
	DBG (Lin et al., 2020)	TSN	57.8	49.4	39.8	30.2	21.7
	G-TAD (Xu, Zhao, Rojas, Thabet, & Ghanem, 2020)	TSN	54.5	47.6	40.3	30.8	23.4
	BC-GNN (Bai et al., 2020)	TSN	57.1	49.1	40.4	31.2	23.1
	TAL-MR (Zhao et al., 2020)	I3D	53.9	50.7	45.4	38.0	28.5
	BMN-CAS (Sridhar et al., 2021)	TSN	64.4	58.0	49.2	38.2	27.8
	RTD-Net (Tan et al., 2021)	I3D	68.3	62.3	51.9	38.8	23.7
	BackTAL (Yang et al., 2021)	I3D	54.4	45.5	36.3	26.2	14.8
	BSN++ (Su et al., 2021)	TSN	59.9	49.5	41.3	31.9	22.8
	MUSES (Liu, Hu, et al., 2021)	I3D	68.9	64.0	56.9	46.3	31.0
	TadTR (Liu, Wang, et al., 2021)	I3D	62.4	57.4	49.2	37.8	26.3
	TCANet (Qing et al., 2021)	TSN	60.6	53.2	44.6	36.8	26.7
	BCNet (Yang et al., 2022)	TSN	66.5	60.0	51.6	41.0	29.2
	BCNet (Yang et al., 2022)	I3D	71.5	67.0	60.0	48.9	33.0
	RCL (Wang et al., 2022)	TSN	70.1	62.3	52.9	42.7	30.7
Anchor-based	R-C3D (Xu et al., 2017)	C3D	44.8	35.6	28.9	–	–
	TAL-Net (Chao et al., 2018)	I3D	53.2	48.5	42.8	33.8	23.4
	GTAN (Long et al., 2019)	P3D	57.8	47.2	38.8	–	–
	PBRNet (Liu & Wang, 2020)	I3D	58.5	54.6	51.3	41.8	29.5
	PCG-TAL (Su, Xu, Sheng, & Ouyang, 2020)	I3D	65.1	59.5	51.2	–	–
Anchor-free	A^2Net (Yang, Peng, Zhang, Fu, & Han, 2020)	I3D	58.6	54.1	45.5	32.5	17.2
	AFSD (Lin et al., 2021)	I3D	67.3	62.4	55.5	43.7	31.1
	TRA (Zhao et al., 2022)	I3D	70.0	64.3	57.4	46.2	31.1
	Actionformer (Zhang et al., 2022)	I3D	75.5	72.5	65.6	56.6	42.7
	Actionformer (Zhang et al., 2022)	TSP	69.5	63.8	56.3	44.8	30.8
	Actionformer (Our reproduce)	I3D	82.1	77.8	71.0	59.4	43.9
	Ours	I3D	82.6	79.1	72.5	59.5	45.4

et al., 2022). Thus, our PBBNet obtains the third-place performance in Table 2. There are only two actionness-based methods (TCANet (Qing et al., 2021) and RCL (Wang et al., 2022)) better than our PBBNet at average mAP. Both TCANet (Qing et al., 2021) and RCL (Wang et al., 2022) use dense anchors to predict action instance with a recurrent refinement method. But our method still outperforms TCANet (Qing et al., 2021) and RCL (Wang et al., 2022) by 1.4% and 0.5% mAP at tIoU=0.5, respectively.

Performance Comparison on HACS Dataset Since HACS is a new-formed large-scale dataset, there are a small number of experimental results reported by existing TAL methods. We evaluate our PBBNet and Actionformer (Zhang et al., 2022) on HACS Dataset and compare the performance between our PBBNet and some TAL methods (Zhao, Xiong, et al., 2017; Zhang et al., 2019; Lin et al., 2019; Xu et al., 2020; Liu, Wang, et al., 2021; Qing et al., 2021; Liu et al., 2022; Zhang et al., 2022). The results are shown in Table 3. Our methods can achieve second-place performance with 38.26% average mAP. In anchor-free methods, we observe that our PBBNet obtain 1.75% average mAP improvement over Actionformer (Zhang et al., 2022) on the large-scale dataset. We assume that our method will benefit from more training data. In actionness-based methods, BMN (Lin et al., 2019) presets dense anchors to generate a large number of action proposals and find out some of them with high actionness scores. With high-quality action proposals from BMN (Lin et al., 2019), TCANet (Qing et al., 2021) can perform exciting performances. In Table 3, we reproduce TCANet (Qing et al., 2021) using open source codes and pre-trained model weights. According to the reproduce results, our PBBNet can achieve comparable performances. Especially, our method has 1.38% mAP improvement at tIoU=0.5 compared with TCANet (Qing et al., 2021).

4.4. Ablation study

To systematically evaluate the effectiveness of our method, we construct the ablation studies of our proposed PPBNet on ActivityNet-v1.3 dataset.

Table 2
Performance comparison between our proposed PPBNet and other SOTA temporal action localization methods on ActivityNet-v1.3 dataset in terms of average mAP. "Average" indicates the average mAP under tIoU thresholds from 0.5 to 0.95 with step size of 0.05.

	Method	Feature	mAP at different tIoUs			
			0.50	0.75	0.95	Average
Others	P-GCN (Zeng et al., 2019)	I3D	48.3	33.2	3.3	31.1
	ContextLoc (Zhu et al., 2021)	I3D	56.0	35.2	3.6	34.2
	VSGN (Zhao, Thabet, & Ghanem, 2021)	TSN	52.4	36.0	8.4	35.1
	VSGN (Zhao, Thabet, & Ghanem, 2021)	I3D	52.3	35.2	8.3	34.7
	VSGN (Zhao, Thabet, & Ghanem, 2021)	TSP	53.3	36.8	8.1	35.9
	DCAN (Chen et al., 2022)	TSN	51.8	36.0	9.5	35.5
	E2E-TAD (Liu et al., 2022)	I3D	49.6	35.2	9.9	34.4
	E2E-TAD (Liu et al., 2022)	Slowfast	50.1	35.8	10.5	35.1
Anctionness-based	CDC (Shou et al., 2017)	C3D	45.3	26.0	0.2	23.8
	BSN (Lin et al., 2018)	TSN	46.5	30.0	8.0	30.0
	BMN (Lin et al., 2019)	TSN	50.1	34.8	8.3	33.9
	G-TAD (Xu et al., 2020)	TSN	50.4	34.6	9.0	34.1
	G-TAD (Xu et al., 2020)	TSP	51.3	37.1	9.3	35.8
	BC-GNN (Bai et al., 2020)	TSN	50.6	34.8	9.4	34.3
	TAL-MR (Zhao et al., 2020)	I3D	43.5	33.9	9.2	30.2
	BSN++ (Su et al., 2021)	TSN	51.3	35.7	8.3	34.9
	RTD-Net (Tan et al., 2021)	I3D	47.2	30.7	8.6	30.8
	MUSES (Liu, Hu, et al., 2021)	I3D	50.0	35.0	6.6	34.0
	TadTR (Liu, Wang, et al., 2021)	I3D	49.1	32.6	8.5	32.3
	TCANet (Qing et al., 2021)	TSN	52.3	36.7	6.9	35.5
	TCANet (Qing et al., 2021)	Slowfast	54.3	39.1	8.4	37.6
	BMN-CAS (Sridhar et al., 2021)	TSN	52.4	36.2	5.2	35.4
	BCNet (Yang et al., 2022)	TSN	53.2	36.2	10.6	35.5
	RCL (Wang et al., 2022)	TSN	51.5	35.3	8.0	34.4
	RCL (Wang et al., 2022)	I3D	54.2	36.2	9.2	36.0
	RCL (Wang et al., 2022)	TSP	55.2	36.2	8.3	37.7
Anchor-based	R-C3D (Xu et al., 2017)	C3D	26.8	–	–	–
	TAL-Net (Chao et al., 2018)	I3D	38.2	18.3	1.3	20.2
	GTAN (Long et al., 2019)	P3D	52.6	34.1	8.9	34.3
	PBRNet (Liu & Wang, 2020)	I3D	54.0	35.0	9.0	35.0
	PCG-TAL (Su et al., 2020)	I3D	44.3	29.9	5.5	28.9
Anchor-free	A^{2}Net (Yang et al., 2020)	I3D	43.6	28.7	3.7	27.8
	AFSD (Lin et al., 2021)	I3D	52.4	35.3	6.5	34.4
	TRA (Zhao et al., 2022)	I3D	52.4	35.1	7.2	34.4
	Actionformer (Zhang et al., 2022)	I3D	53.5	36.2	8.2	35.6
	Actionformer (Zhang et al., 2022)	TSP	54.1	36.3	7.7	36.0
	Ours	TSP	55.7	38.2	7.5	37.1

Table 3
Performance comparison between our proposed PPBNet and other SOTA temporal action localization methods on HACS dataset in terms of average mAP.

	Method	Feature	mAP at different tIoUs			
			0.50	0.75	0.95	Average
Actionness-based	SSN (Zhao, Xiong, et al., 2017)	TSN	28.82	18.80	5.32	18.97
	2019-Winner (Zhang, Peng, Yang, Fu, & Luo, 2019)	Slowfast	–	–	–	23.49
	BMN (Lin et al., 2019)	Slowfast	52.49	36.38	10.37	35.76
	G-TAD (Xu et al., 2020)	I3D	41.08	27.59	8.34	27.48
	TadTR (Liu, Wang, et al., 2021)	I3D	45.16	30.70	11.78	30.83
	TadTR (Liu, Wang, et al., 2021)	TSM	30.69	18.94	5.26	18.28
	TCANet[SW] (Qing et al., 2021)	Slowfast	54.14	37.24	11.32	36.79
	TCANet[BMN] (Qing et al., 2021)	Slowfast	56.74	41.14	12.15	39.77
	TCANet[BMN] (our reproduce)	Slowfast	55.30	39.47	11.65	38.54
	E2E-TAD[TadTR] (Liu et al., 2022)	I3D	40.32	24.97	7.71	25.70
Anchor-free	*Actionformer (Zhang et al., 2022)	Slowfast	54.53	36.94	10.86	36.51
	Ours	Slowfast	56.68	38.66	11.42	38.26

Impact of components in PBBNet In Table 4, we compare ablation models of our PBBNet to demonstrate the usefulness of each modules, where "IBM", "TCM" and "FPBM" indicate the instance-wise boundary-aware module, temporal context-aware module and frame-wise progressive boundary-aware module, respectively. It should be noted that we use 1D temporal convolution operations to capture temporal context information like AFSD (Lin et al., 2021) if no using temporal context-aware module (TCM). Our baseline method is the PPBNet model only using IBM, which achieves 36.25% average mAP. "PPBNet w/o FPBM" means that we use PPBNet model without FPBM to detect action instances, which obtain 0.25% improvement over the baseline method on average mAP. It

Table 4
Ablation study of our PBBNet components on ActivityNet-v1.3 dataset in terms of average mAP. "IBM" indicates Instance-wise Boundary-aware Module, "TCM" indicates Temporal Context-aware Module and "FPBM" indicates Frame-wise Progressive Boundary-aware Module.

Methods	IBM	TCM	FPBM	mAP at different tIoUs			
				0.5	0.75	0.95	Average
Baseline	✓			54.56	37.22	7.85	36.25
PBBNet w/o FPBM	✓	✓		54.75	37.59	7.84	36.45
PBBNet w/o TCM	✓		✓	55.30	37.69	7.32	36.53
PBBNet	✓	✓	✓	55.66	38.16	7.47	37.05

Table 5
Ablation study about the number of pyramid layers in instance-wise boundary-aware module on ActivityNet-v1.3 dataset.

N_L	Input	Output	mAP at different tIoUs			
			0.50	0.75	0.95	Average
0	$F \in \mathbb{R}^{T\times C}$	$\{F^{(0)}\},\ F^{(0)} \in \mathbb{R}^{T\times C}$	52.84	36.42	7.33	35.29
2	$F \in \mathbb{R}^{T\times C}$	$\{F^{(0)}, F^{(1)}, F^{(2)}\}, F^{(N_i)} \in \mathbb{R}^{\frac{T}{2^i}\times C}$	54.11	37.15	7.42	35.93
5	$F \in \mathbb{R}^{T\times C}$	$\{F^{(0)}, F^{(1)}, \dots, F^{(N_6)}\}, F^{(N_i)} \in \mathbb{R}^{\frac{T}{2^i}\times C}$	55.66	38.16	7.47	37.05

Table 6
Ablation study about the progressive boundary-aware boosting strategy on ActivityNet-v1.3 dataset.

ID	Methods	Supervision	mAP at different tIoUs			
			0.50	0.75	0.95	Average
1	PBBNet w/o boundary-aware boosting	–	54.75	37.59	7.84	36.45
2	PBBNet w/ 1-step boundary-aware boosting	tIoU = 0.5	55.39	38.06	7.65	36.80
3	PBBNet w/ 1-step boundary-aware boosting	tIoU = 0.6	55.29	38.00	7.93	36.85
4	PBBNet w/ 1-step boundary-aware boosting	tIoU = 0.7	55.39	38.01	8.17	36.88
5	PBBNet w/ 1-step boundary-aware boosting	tIoU = 0.8	55.29	37.74	8.36	36.75
6	PBBNet w/ 1-step boundary-aware boosting	tIoU = 0.9	55.26	37.59	8.06	36.60
7	PBBNet w/ 2-step boundary-aware boosting	tIoUs = [0.5,0.6]	55.53	38.05	7.66	36.91
8	PBBNet w/ 2-step boundary-aware boosting	tIoUs = [0.5,0.7]	55.76	38.03	7.67	36.99
9	PBBNet w/ 2-step boundary-aware boosting	tIoUs = [0.5,0.8]	55.51	38.04	7.92	36.91
10	PBBNet w/ 2-step boundary-aware boosting	tIoUs = [0.5,0.9]	55.45	38.08	7.86	36.88
11	PBBNet w/ 3-step boundary-aware boosting	tIoUs = [0.5,0.5,0.5]	55.73	37.68	6.70	36.67
12	PBBNet w/ 3-step boundary-aware boosting	tIoUs = [0.7,0.7,0.7]	55.53	38.19	7.72	36.94
13	PBBNet w/ 3-step boundary-aware boosting	tIoUs = [0.9,0.9,0.9]	55.38	37.77	8.56	36.87
14	PBBNet w/ 3-step boundary-aware boosting	tIoUs = [0.5,0.6,0.7]	55.66	38.16	7.47	**37.05**
15	PBBNet w/ 3-step boundary-aware boosting	tIoUs = [0.5,0.7,0.9]	55.59	38.14	7.88	37.02

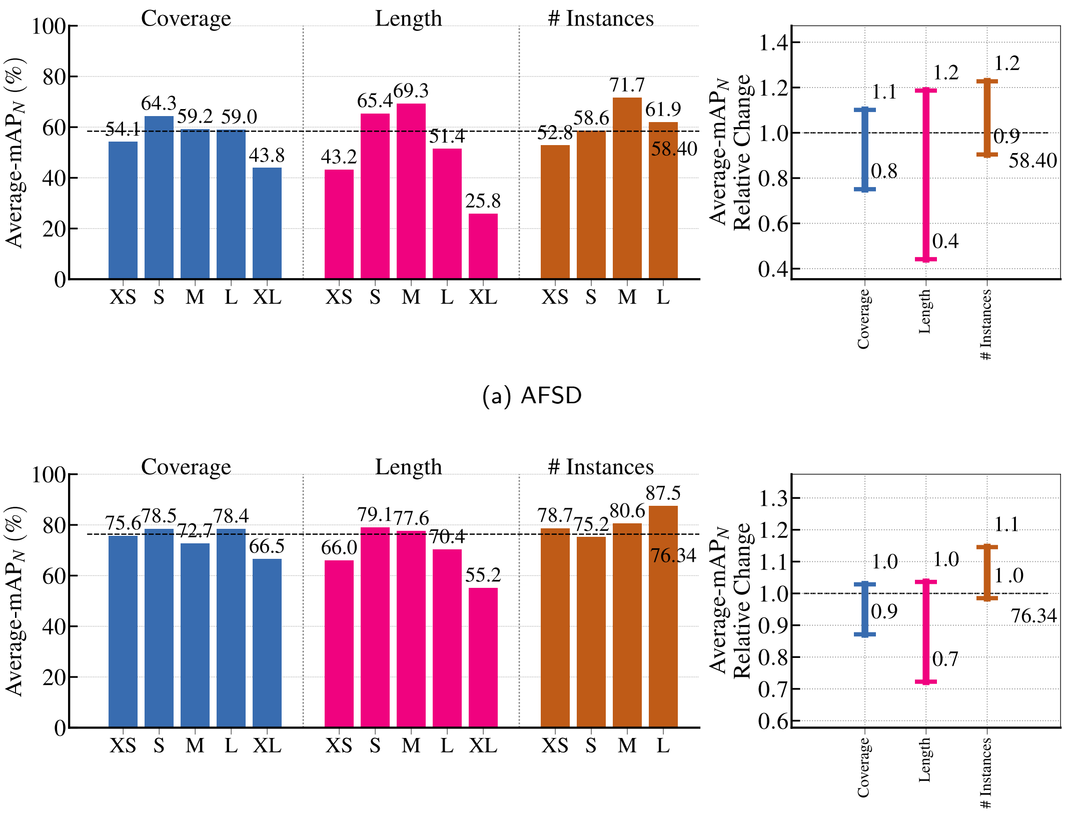
Table 7
Results of our PBBNet at different stages on ActivityNet-v1.3 dataset. The 3-step progressive refinement method and supervision information tIoUs = [0.5,0.6,0.7] are used.

Stage	mAP at different tIoUs			
	0.50	0.75	0.95	Average
IBM stage	54.75	37.59	7.84	36.45
FPBM (1st step) stage	55.58	38.16	7.69	36.93
FPBM (2nd step) stage	55.65	38.15	7.41	37.02
FPBM (3rd step) stage	55.66	38.16	7.47	37.05

can prove that our TCM has a stronger ability to utilize temporal context information compared with 1D temporal convolutions. "PPBNet w/o TCM" means that we combine IBM and FPBM to localize action instances, which outperforms our baseline by 0.28% on average mAP. It shows the effectiveness of our FPBM. With all the modules including IBM, TCM and FPBM, PBBNet achieves 0.85% average mAP improvement over our baseline method. Noteworthy, PBBNet outperforms "PPBNet w/o TCM" by 0.52% on average mAP. It proves the effectiveness of our TCM again, especially in our progressive coarse-to-fine framework.

Impact of the number of pyramid layers The pyramid network of instance-wise boundary-aware module is used to generate multi-scale video features to predict action instances at different scales. N_T is the number of pyramid layers in instance-wise boundary-aware module. In Table 5, we explore the effect of N_T on ActivityNet-v1.3 dataset. It can be seen that our PPBNet obtains the best performances when $N_T = 5$. Thus, we set N_T to 5. We also try to interact the features of adjacent pyramid layers likes (Kim, Kook, Sun, Kang, & Ko, 2018), but it offers little performance improvement.

Impact of progressive boundary-aware boosting strategy To better evaluate the effectiveness of our frame-wise progressive boundary-aware module, in Table 6, we construct a ablation study to explore the effect of different progressive boundary-aware



(a) AFSD

(b) Ours

Fig. 2. Sensitivity analysis of AFSD (Lin et al., 2021) and our PBBNet on THUMOS14 dataset.

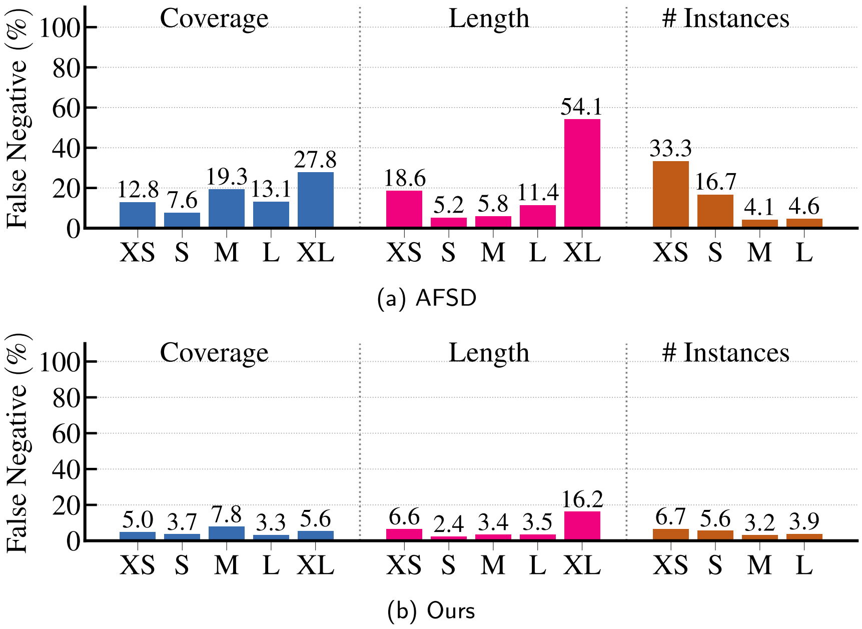
boosting strategies. From Table 6, we can obtain the following observations: (1) The performance of all models with boundary-aware boosting is higher than the model without boundary-aware boosting. (2) Using boundary-aware boosting with more steps, our PBBNet models can obtain better results. For example, the id-14 model using 3-step boundary-aware boosting and tIoUs = [0.5,0.6,0.7] outperforms the id-6 model using 1-step boundary-aware boosting and tIoUs = [0.7] by 0.45% average mAP. (3) Focusing on ids from 11 to 15, models using supervision information from weak to strong achieve better results compared with models using invariable supervision information. For example, the id-14 model using tIoUs=[0.5,0.6,0.7] as supervision performs 0.38% average mAP improvement over the id-11 model with supervision of tIoUs = [0.5,0.5,0.5]. (4) As the number of boundary-aware boosting increases, the performance improvement from each additional boundary-aware boosting decreases. For example, comparing the best models, the performance improvement is 0.43% average mAP with the number of steps from 0 to 1, while the performance improvement is 0.06% average mAP with the number of steps from 2 to 3. Based on these observations, we finally adopt the 3-step boundary-aware boosting strategy with supervision of tIoUs = [0.5,0.6,0.7], i.e., the id-14 model in Table 6.

Results of PBBNet at different stages

Table 7 shows the results of our PBBNet at different stages, where the 3-step progressive refinement method and supervision information tIoUs = [0.5,0.6,0.7] are used. It can be seen that the performances of our PBBNet are improved stage by stage with the progressive boundary-aware boosting. , temporal context-aware module and frame-wise progressive boundary-aware module The results of IBM stage means the predictions from the instance-wise boundary-aware module. The results of FPBM (1st step), FPBM (2nd step) and FPBM (3rd step) are corresponding to three predictions from the 3-step frame-wise progressive boundary-aware module, respectively. Interestingly, the results of FPBM (1st step) and FPBM (2nd step) are higher than the 1-step and 2-step boundary-aware models that are in Table 6, respectively.

4.5. Analysis

Following previous works (Zhang et al., 2022; Wang et al., 2022), we evaluate our method using a temporal action localization analysis tool named DETAD (Alwassel, Heilbron, Escorcia, & Ghanem, 2018). In the DETAD, three characteristic measurements are designed to evaluate the comprehensive performance of TAL models, including coverage, length, and instance. To be specific, the coverage metric focuses on the ratio of action duration to the duration of the whole video, which splits untrimmed videos into



(a) AFSD

(b) Ours

Fig. 3. False negative analysis of AFSD (Lin et al., 2021) and our PBBNet on THUMOS14 dataset.

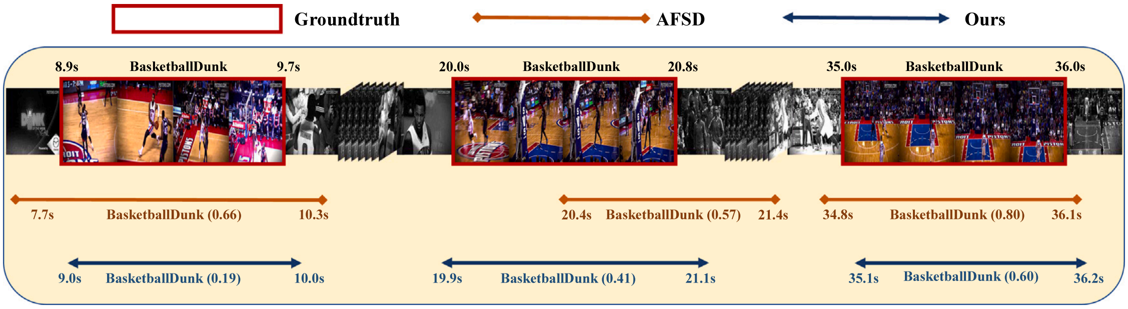


Fig. 4. Visualization of top-3 results generated by AFSD (Lin et al., 2021) and our PBBNet on THUMOS14 dataset.

extra small (XS), small (S), medium (M), large (L), and extra large (XL) with multiple thresholds [0.02, 0.04, 0.06, 0.08]. Unlike the coverage metric, the length metric classifies videos into five categories using the absolute duration of action instances with a series of duration thresholds [3 s, 6 s, 12 s, 18 s]. Furthermore, the instance metric categorizes videos by the number of action instances they contain, where the thresholds are set to [2, 40, 80].

For sensitivity analysis, we test the performances of AFSD (Lin et al., 2021) and our PBBNet under different types of untrimmed videos that are categorized by the metric of coverage, length or instance number. As shown in Fig. 2, the comprehensive performance of our PPBNet outperforms AFSD by 17.94% average mAP. Moreover, the relative change of our PBBNet is smaller than AFSD in each characteristic evaluation. We also analyze the false negative of AFSD (Lin et al., 2021) and our PBBNet in the same way. In Fig. 3, it can be seen that the false negative rates of our PPBNet are all lower than AFSD. In particular, our method reduces the false negative rate of untrimmed videos that contain action instances with extra large lengths from 54.1% to 16.2%.

To further analyze the predicted results, we show the top 3 action instances predicted by AFSD (Lin et al., 2021) and our PBBNet using THUMOS14 dataset in Fig. 4. The video named "0000560" is used, which contains 3 action instances (groundtruth). It can be seen that both AFSD and our PPBNet predict the right class of each action instance. However, when predicting the boundaries of action instance continuing from 20.0s to 20.8s, the result of AFSD deviates significantly from the groundtruth. The action instance is predicted to continue from 20.4s to 21.4s by AFSD with a confidence score 0.57. Compared with AFSD, our PBBNet generates a better result (from 19.9s to 21.1s). Besides, our method produces a more accurate boundary result than AFSD for the action instance that continues from 8.9s to 9.7s. These demonstrate the effectiveness of our method in boosting boundary predictions.

5. Conclusion

This paper introduced a Progressive Boundary-aware Boosting Network (PBBNet) for anchor-free temporal action localization, which could predict high-quality action predictions. We focused on the inaccurate action boundary predictions of anchor-free methods and designed an instance-wise boundary-aware module and frame-wise progressive boundary-aware module to boost the boundary predictions. Besides, we developed a temporal context-aware module to capture temporal context information, which helped our model obtain better results. Extensive experiments on THUMOS14, ActivityNet-v1.3, and HACS datasets proved the effectiveness of our PBBNet. Our method outperforms all existing anchor-free models in temporal action localization. In particular, the PBBNet achieves state-of-the-art performance on THUMOS14 dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Beijing Natural Science Foundation (JQ20022); the National Natural Science Foundation of China (62072026, 62102419, 72171093); the Fundamental Research Funds for the Central Universities (2022JBMC054); the open research fund of the State Key Laboratory for Management and Control of Complex Systems (20220103); the open project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University (MMC202103); the Key Laboratory Foundation of Information Perception and Systems for Public Security of MIIT (Nanjing University of Science and Technology) under Grant 202005; the Chinese Association for Artificial Intelligence (CAAI)-Huawei MindSpore Open Fund. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and the Ascend AI Processor used for this research. The codes using MindSpore (https://www.mindspore.cn) will also be released at https://github.com/cjzhang7/bjtu-mepro-anchorfree.

References

Alkanat, T., Akdag, E., Bondarev, E., & de With, P. H. (2022). Density-guided label smoothing for temporal localization of driving actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops* (pp. 3174–3182).

Alwassel, H., Giancola, S., & Ghanem, B. (2021). TSP: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE international conference on computer vision* (pp. 3173–3183).

Alwassel, H., Heilbron, F. C., Escorcia, V., & Ghanem, B. (2018). Diagnosing error in temporal action detectors. In *Proceedings of the European conference on computer vision* (pp. 256–272).

Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., & Liu, J. (2020). Boundary content graph neural network for temporal action proposal generation. In *Proceedings of the European conference on computer vision* (pp. 121–137). Springer.

Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 961–970).

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6299–6308).

Chao, Y.-W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., & Sukthankar, R. (2018). Rethinking the faster R-CNN architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1130–1139).

Chen, P., Gan, C., Shen, G., Huang, W., Zeng, R., & Tan, M. (2019). Relation attention for temporal action localization. *IEEE Transactions on Multimedia*, *22*(10), 2723–2733.

Chen, G., Zheng, Y.-D., Wang, L., & Lu, T. (2022). DCAN: Improving temporal action detection via dual context aggregation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 1* (pp. 248–257).

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., et al. (2020). Rethinking attention with performers. arXiv preprint arXiv:2009.14794.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 764–773).

Dave, I., Scheffer, Z., Kumar, A., Shiraz, S., Rawat, Y. S., & Shah, M. (2022). GabriellaV2: Towards better generalization in surveillance videos for action detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV) workshops* (pp. 122–132).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 6202–6211).

Gao, J., Shi, Z., Wang, G., Li, J., Yuan, Y., Ge, S., et al. (2020). Accurate temporal action proposal generation with relation-aware pyramid network. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07* (pp. 10810–10817).

Hassani, H., Ershadi, M. J., & Mohebi, A. (2022). LVTIA: A new method for keyphrase extraction from scientific video lectures. *Information Processing & Management*, *59*(2), Article 102802. http://dx.doi.org/10.1016/j.ipm.2021.102802, URL: https://www.sciencedirect.com/science/article/pii/S030645732100279X.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

Hosono, T., Sawada, K., Sun, Y., Hayase, K., & Shimamura, J. (2020). Activity normalization for activity detection in surveillance videos. In *2020 IEEE international conference on image processing* (pp. 1386–1390). http://dx.doi.org/10.1109/ICIP40778.2020.9190884.

Jiang, Y.-G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., et al. (2014). THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Kim, S.-W., Kook, H.-K., Sun, J.-Y., Kang, M.-C., & Ko, S.-J. (2018). Parallel feature pyramid network for object detection. In *Proceedings of the European conference on computer vision* (pp. 234–250).

Li, W., Chen, S., Gu, J., Wang, N., Chen, C., & Guo, Y. (2022). MV-TAL: Mulit-view temporal action localization in naturalistic driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops* (pp. 3242–3248).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., et al. (2020). Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11499–11506).

Lin, T., Liu, X., Li, X., Ding, E., & Wen, S. (2019). BMN: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3889–3898).

Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., et al. (2021). Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3320–3329).

Lin, T., Zhao, X., & Shou, Z. (2017). Single shot temporal action detection. In *Proceedings of the ACM international conference on multimedia* (pp. 988–996).

Lin, T., Zhao, X., Su, H., Wang, C., & Yang, M. (2018). BSN: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision* (pp. 3–19).

Liu, X., Bai, S., & Bai, X. (2022). An empirical study of end-to-end temporal action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 20010–20019).

Liu, X., Hu, Y., Bai, S., Ding, F., Bai, X., & Torr, P. H. (2021). Multi-shot temporal event localization: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 12596–12606).

Liu, Y., Ma, L., Zhang, Y., Liu, W., & Chang, S.-F. (2019). Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3604–3613).

Liu, Q., & Wang, Z. (2020). Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07* (pp. 11612–11619).

Liu, X., Wang, Q., Hu, Y., Tang, X., Bai, S., & Bai, X. (2021). End-to-end temporal action detection with transformer. arXiv preprint arXiv:2106.10271.

Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., & Mei, T. (2019). Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 344–353).

Nie, J., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., & Shao, L. (2019). Enriched feature guided refinement network for object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 9537–9546).

Pan, Y., Li, Z., Zhang, L., & Tang, J. (2021). Distilling knowledge in causal inference for unbiased visual question answering. In *Proceedings of the 2nd ACM international conference on multimedia in Asia* (pp. 1–7).

Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., et al. (2021). Temporal context aggregation network for temporal action proposal refinement. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 485–494).

Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3D residual networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5533–5541).

Rani, S., & Kumar, M. (2020). Social media video summarization using multi-visual features and Kohnen's Self Organizing Map. *Information Processing & Management*, *57*(3), Article 102190. http://dx.doi.org/10.1016/j.ipm.2019.102190, URL: https://www.sciencedirect.com/science/article/pii/S0306457319308556.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the advances in neural information processing systems, vol. 28*.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 658–666).

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

Shou, Z., Wang, D., & Chang, S.-F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1049–1058).

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proceedings of the advances in neural information processing systems, vol. 27*.

Sridhar, D., Quader, N., Muralidharan, S., Li, Y., Dai, P., & Lu, J. (2021). Class semantics-based attention for action detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 13739–13748).

Su, H., Gan, W., Wu, W., Qiao, Y., & Yan, J. (2021). BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 3* (pp. 2602–2610).

Su, R., Xu, D., Sheng, L., & Ouyang, W. (2020). PCG-TAL: Progressive cross-granularity cooperation for temporal action localization. *IEEE Transactions on Image Processing*, *30*, 2103–2113.

Tan, J., Tang, J., Wang, L., & Wu, G. (2021). Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 13526–13535).

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021). Going deeper with image transformers. In *Proceedings of the IEEE international conference on computer vision* (pp. 32–42).

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489–4497).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems, vol. 30*.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, Q., Zhang, Y., Zheng, Y., & Pan, P. (2022). RCL: Recurrent continuous localization for temporal action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13566–13575).

Wu, Y., Zhang, K., Wu, D., Wang, C., Yuan, C.-A., Qin, X., et al. (2021). Person reidentification by multiscale feature representation learning with random batch feature mask. *IEEE Transactions on Cognitive and Developmental Systems*, *13*(4), 865–874. http://dx.doi.org/10.1109/TCDS.2020.3003674.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Xu, H., Das, A., & Saenko, K. (2017). R-C3D: Region convolutional 3D network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 5783–5792).

Xu, M., Zhao, C., Rojas, D. S., Thabet, A., & Ghanem, B. (2020). G-TAD: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10156–10165).

Yang, L., Han, J., Zhao, T., Lin, T., Zhang, D., & Chen, J. (2021). Background-click supervision for temporal action localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yang, L., Peng, H., Zhang, D., Fu, J., & Han, J. (2020). Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, *29*, 8535–8548.

Yang, H., Wu, W., Wang, L., Jin, S., Xia, B., Yao, H., et al. (2022). Temporal action proposal generation with background constraint. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 3* (pp. 3054–3062).

Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., et al. (2019). Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 7094–7103).

Zhang, Z., Jiang, W., Qin, J., Zhang, L., Li, F., Zhang, M., et al. (2017). Jointly learning structured analysis discriminative dictionary and analysis multiclass classifier. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3798–3814.

Zhang, Z., Li, F., Jia, L., Qin, J., Zhang, L., & Yan, S. (2017). Robust adaptive embedded label propagation with weight learning for inductive classification. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(8), 3388–3403.

Zhang, Z., Li, F., Zhao, M., Zhang, L., & Yan, S. (2016). Joint low-rank and sparse principal feature coding for enhanced robust representation and visual classification. *IEEE Transactions on Image Processing*, *25*(6), 2429–2443.

Zhang, S., Peng, H., Yang, L., Fu, J., & Luo, J. (2019). Learning sparse 2D temporal adjacent networks for temporal action localization. arXiv preprint arXiv:1912.03612.

Zhang, C., Wu, J., & Li, Y. (2022). Actionformer: Localizing moments of actions with transformers. arXiv preprint arXiv:2202.07925.

Zhao, C., Thabet, A. K., & Ghanem, B. (2021). Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 13658–13667).

Zhao, H., Torralba, A., Torresani, L., & Yan, Z. (2019). Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 8668–8678).

Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., & Tian, Q. (2020). Bottom-up temporal action localization with mutual regularization. In *Proceedings of the European conference on computer vision* (pp. 539–555). Springer.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D. (2017). Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2914–2923).

Zhao, Y., Zhang, H., Gao, Z., Guan, W., Nie, J., Liu, A., et al. (2022). A temporal-aware relation and attention network for temporal action localization. *IEEE Transactions on Image Processing*, *31*, 4746–4760. http://dx.doi.org/10.1109/TIP.2022.3182866.

Zhao, G., Zhang, M., Li, Y., Liu, J., Zhang, B., & Wen, J.-R. (2021). Pyramid regional graph representation learning for content-based video retrieval. *Information Processing & Management*, *58*(3), Article 102488. http://dx.doi.org/10.1016/j.ipm.2020.102488, URL: https://www.sciencedirect.com/science/article/pii/S0306457320309766.

Zhao, Y., Zhang, B., Wu, Z., Yang, S., Zhou, L., Yan, S., et al. (2017). Cuhk & ethz & siat submission to activitynet challenge 2017. *8*(8), arXiv preprint arXiv:1710.08011.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07* (pp. 12993–13000).

Zhu, Z., Tang, W., Wang, L., Zheng, N., & Hua, G. (2021). Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 13516–13525).