MR.CAN: CLASS-AWARE NETWORK WITH MULTI-RELATIONS FOR TEMPORAL ACTION DETECTION

*Dongqi Wang, Zixuan Zhao and Xu Zhao**

Department of Automation, Shanghai Jiao Tong University, Shanghai, China

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

ABSTRACT

Recognizing action patterns and exploring multiple relations are vital for Temporal Action Detection (TAD) task, which aims at locating and classifying action segments in untrimmed videos. However, most existing methods attempt to build a general model to handle diverse actions, ignoring the huge difference between various classes. Besides, the exploration of temporal and semantic relations between different segments remains an ongoing challenge due to complex video content. In this paper, we contend that different action classes should be processed differently and thus design a new Class-Aware Mechanism to achieve accurate detection. Moreover, an effective module named Multi-relations Builder is proposed to establish temporal and semantic relations simultaneously. These two modules are integrated as Class-Aware Network with Multi-relations (MrCAN). In comprehensive experiments conducted on two benchmarks, it outperforms all other current methods and achieves state-of-the-art performance, improving the average mAP from 45.78% to 48.98% on THUMOS-14 and from 35.52% to 35.87% on ActivityNet-1.3 respectively. Furthermore, the well-designed Multi-relations Builder can also be used to boost some other existing methods.

Index Terms— Video Analysis, Temporal Action Detection

1. INTRODUCTION

As a significant task of video understanding, Temporal Action Detection (TAD) aims at locating the boundary of every action instance and classifying its class label in untrimmed videos. Inspired by the success of object detection, two-stage pipeline [1, 2, 3, 4, 5] prevails in TAD: the first stage generates candidate action segments (*i.e.*, proposals), which are then labelled with certain classes in the second stage. Overall, performance of TAD largely depends on two aspects. Firstly, recognizing action patterns bridges semantic information with video representations. Secondly, exploring rich logical relations within videos gives the model a larger receptive field, and also makes the video representation more expressive.

In terms of pattern recognition, as shown in Fig.1, videos are firstly divided into several snippets and encoded as feature sequence. Next, some methods [1, 3] try to recognize boundary patterns of each snippet to locate action boundaries. Meanwhile, action patterns within predefined anchors can be recognized to evaluate confidence score of each anchor [2, 4, 6]. However, almost all methods [1, 2, 3, 4, 6] force the model to cater for all kinds of actions, which means that a universal pattern has to be summarized to locate every action's start, end and actionness. It has great limitations

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

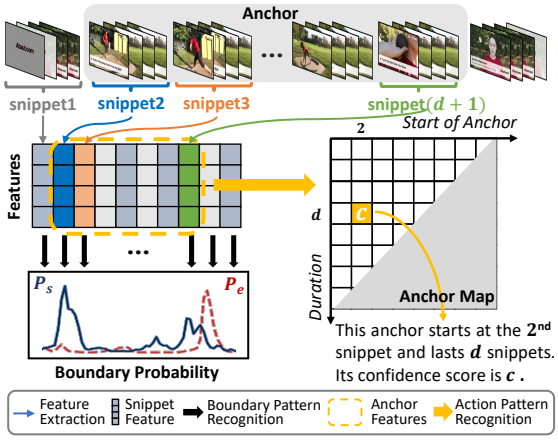


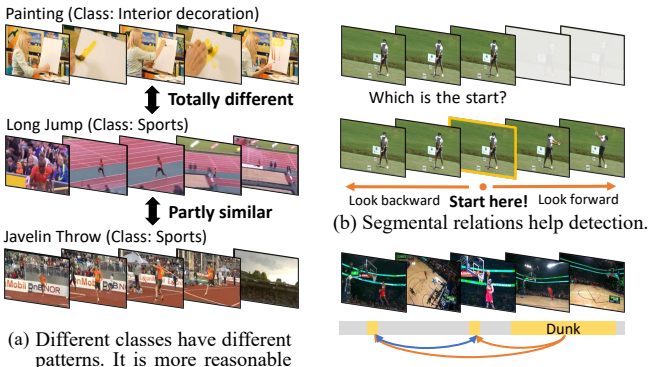
Fig. 1: Pattern recognition in TAD. Model locates the start and end time by boundary pattern of each snippet, and evaluate anchor's confidence by action pattern of each anchor.

because patterns vary dramatically with action classes. As shown in Fig.2(a), *Painting* (of decoration class) and *Long Jump* (of sports class) are completely different in scenes, motion amplitudes, and frequencies; but the *Javelin Throwing*, which also belongs to sports, is similar to *Long Jump*: for example, both are large amplitudes movements that occur in stadiums and start with running. Rather than existing general models, it is more reasonable to roughly classify actions first and then detect them distinctively based on class.

As for exploring relations in videos, 1D convolution is widely applied to process local temporal relations [1, 2, 3, 6]. Graph convolution network is used to model relations between snippets [4, 7] or segments [5]. Recently, transformer [8] is adopted to establish semantic relations between distant segments. In fact, complex relations in video pose an important challenge to accurate detection. As illustrated in Fig.2(b), some boundaries are very vague, and it is difficult to tell actions from backgrounds based on limited local information. However, if the model is allowed to look forward and backward and grasp the segmental temporal relations, detection will be easier. In addition to the local and segmental temporal relations, semantic relations are also crucial. For example, in Fig.2(c), duration of one action can be varied (such as normal actions and slow-motion playback), but they are similar semantically. Although some actions may be hard to probe, if the model has seen an understandable similar action, then semantic relations between them will be beneficial for complete detection. In general, taking all relations into account is critical and this key task still requires further research.

To tackle these two challenges, we propose **Class-Aware Network with Multi-relations (MrCAN)** and there are two crucial designs inside. (1) For unambiguous recognition of diverse actions, we

2022 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-8563-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICME52920.2022.9860007



(a) Different classes have different patterns. It is more reasonable to treat them distinctively.

(b) Segmental relations help detection.

(c) Semantic relations help detection.

Fig. 2: Motivations of this paper. (a) Action patterns vary dramatically with classes. It is more reasonable to detect actions distinctively based on class. (b) Temporal relations and (c) semantic relations are crucial for detection because they endow models with broader view and better understanding of video content.

contend that completely different classes should be processed separately, and some partly similar classes can be treated together. To this end, **Class-Aware Mechanism (CAM)** is introduced. It includes several action branches and a universal branch. Each action branch takes charge of one specified class and the universal branch supplies universal information. After obtaining a sketchy and general class label of raw video from a video-level classifier, corresponding branches in CAM are activated to generate predictions. (2) In order to fully explore video relations at multiple levels simultaneously, **Multi-relations Builder (MULR)** is designed. MULR constructs three paths for different purposes: the first path is composed of 1D convolution to extract local information; the second path uses GRU flexibly to broaden receptive field of the model and explore segmental temporal relations; and the third path contains two cascaded self-attention modules to grasp semantic relations. Experiments have verified that this builder can improve performance and reduce computational cost by 38% than graph model GTAD [4]. Furthermore, as an independent module, MULR can also be generalized to other methods like BMN [2] and TSI [3]. In summary, our work has three main contributions:

1. We contend that different action classes should be treated differently to generate accurate action segments, and thus the Class-Aware Mechanism (CAM) is proposed to implement this insight.

2. Multi-relations Builder (MULR) is designed to explore local&segmental temporal relations and global semantic relations in an unified module. This builder not only plays an important role in this work, but also is effective for some other existing models.

3. We propose the Class-Aware Network with Multi-relations (MrCAN) for TAD task and verify its effectiveness on two benchmarks. MrCAN achieves state-of-the-art performance using a variety of feature extractors. Using the two-stream feature, MrCAN promotes the average mAP from current best 35.52% to 35.87% on ActivityNet-1.3 and from 45.78% to 48.98% on THUMOS-14.

2. METHOD

2.1. Pipeline

Pipeline of MrCAN is shown in Fig.3. Input is video $V = \{v_i\}_{i=1}^{l_v}$, v_i representing the i-th frame of all l_v frames. Ground-truth actions are denoted as $\Phi_g = \{\varphi_i = (t_{s,i}, t_{e,i}, c_i)\}_{i=1}^{N_g}$. Here, N_g is the number of ground truths, with t_s, t_e and c indicating the start time, end time and action label of each instance respectively. Output is predicted segments $\Phi_p = \{\varphi_i = (t_{s,i}, t_{e,i}, c_i, score_i)\}_{i=1}^{N_p}$. N_p is the quantity of predicted actions and $score$ is the confidence score.

Firstly, video V is sent to the first action classifier and feature extractor to obtain the video-level action class θ and feature sequence F_v separately. Next, Multi-relations Builder explores temporal and semantic relations within the feature F_v. After that, in Class-Aware Mechanism, given predicted class θ, *action branches* and the *universal branch* are merged together to generate boundary probabilities of each snippet, as well as IoU score, center offset and duration offset of each anchor. All the output are fused to generate action proposals Φ_p in Proposal Selection Module. Finally, another proposal-level classifier tags every proposal with a fine action label.

2.2. Feature Extractor and Action Classifier

Following previous work [1, 2, 3], two-stream network [9] with architecture designed in [10] serves as feature extractor. Videos can be divided into several snippets at a regular interval of σ frames, as shown in Fig.1. Each snippet is encoded as $f_s \in \mathbb{R}^C$ by concatenating the output score of top fc-layer in two-stream network, then feature sequence of the whole video can be denoted as $F_v = \{f_i\}_{i=1}^{T} \subset \mathbb{R}^{C \times T}$, where $T = l_v/\sigma$. Besides, features provided by [4, 11, 12] are also adopted for fair and comprehensive comparison.

For accurate and reasonable detection, MrCAN roughly classifies videos firstly and then detect them distinctively. A video-level classifier of m classes scans the whole video and predicts a sketchy action class, *e.g.*"Sports&Exercise". And a proposal-level classifier of n classes ($m \leq n$) receives proposals and tags them with finer labels, *e.g.*"Long Jump". For fair comparison, following [1, 2, 3, 13, 14], Untrimmed Net [15] and the model designed by [16] serve as action classifiers. More details of feature extractor and classifiers are elaborated in **Supplementary Materials**.

2.3. Class-Aware Mechanism

In order to process completely different classes separately for accurate detection, Class-Aware Mechanism (CAM) is proposed. As shown in Fig.3, CAM has four blocks with different outputs, and these blocks are instantiated by two basic modules: Boundary Pattern Recognizer (BPR) and Action Pattern Recognizer (APR). Both BPR and APR are equipped with m action branches and a universal branch. Each action branch takes charge of one specified class and learn corresponding patterns unambiguously, while universal branch distills universal information applicable for most classes.

BPR aims at predicting boundary probabilities P_s and P_e of each snippet. Structure of BPR is illustrated in Fig.4(left). Receiving $F_v \in \mathbb{R}^{C \times T}$, Start-BPR generates start probabilities $P_{s,act} \in \mathbb{R}^{m \times T}$ from m action branches and $P_{s,uni} \in \mathbb{R}^{T}$ from universal branch. Similarly, End-BPR produces $P_{e,act}$ and $P_{e,uni}$. In the later Filter&Fusion Module, given the video-level classification results in one-hot form $\theta \in \mathbb{R}^m$, action branches and universal branches are fused through weights $\omega_{BPR} \in \mathbb{R}^m$ ($\mathbf{0} \leq \omega_{BPR} \leq \mathbf{1}$), as shown in Eq.(1)(2). Here, '$\cdot$' represents dot product and '$\odot$' represents Hadamard product. The Filter&Fuse Module filters output of action branches according to class label θ, and fuses action branches with the universal branch via ω_{BPR}. ω_{BPR} is trainable because the accurate detection of some actions needs more exclusive features while others may depend more on universal patterns, and setting ω_{BPR} trainable transfers the decision right to the model itself.

$$P_{s,ff} = (\theta \odot \omega_{BPR}) \cdot P_{s,act} + (1 - \theta \cdot \omega_{BPR}) P_{s,uni} \quad (1)$$

$$P_{e,ff} = (\theta \odot \omega_{BPR}) \cdot P_{e,act} + (1 - \theta \cdot \omega_{BPR}) P_{e,uni} \quad (2)$$

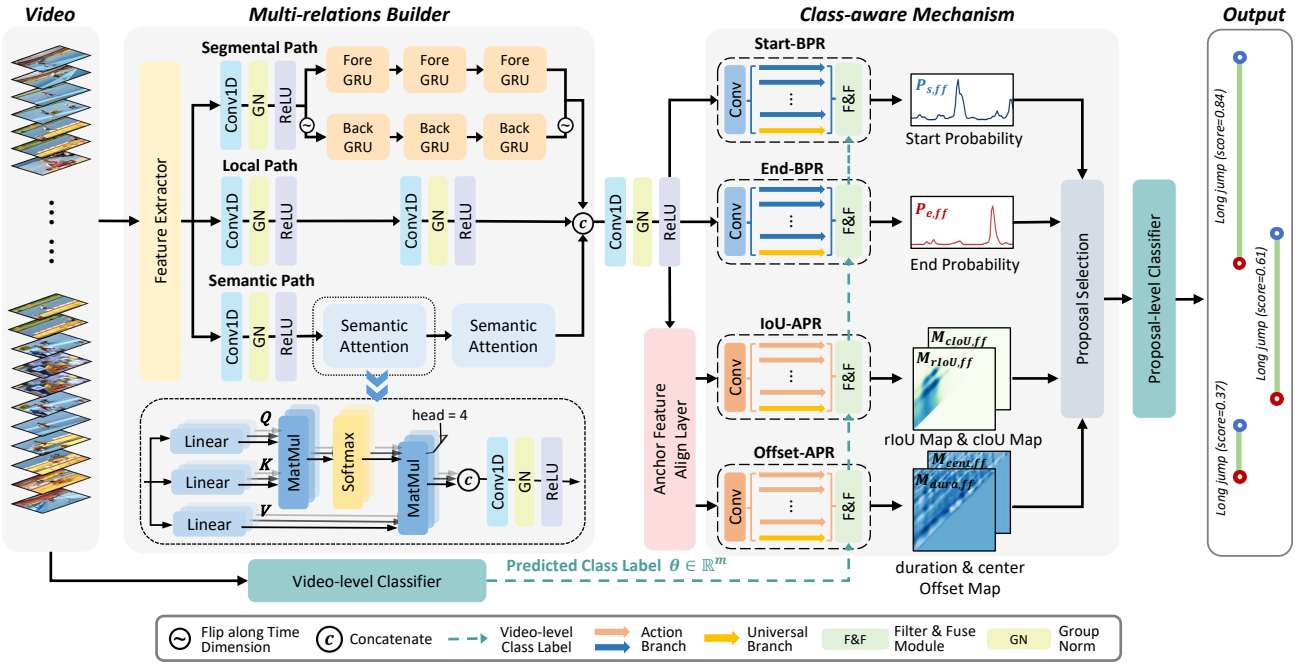


Fig. 3: Overview of MrCAN. Multi-relations Builder explores local temporal relations, segmental temporal relations, and global semantic relations simultaneously. Class-Aware Mechanism processes each class differently based on action branches, and generates boundary probabilities of each snippet, as well as IoU maps and offset maps of each anchor.

APR is designed to recognize action patterns of anchors. Anchor starts at the s-th snippet and lasts d snippets is denoted as (s, d). As shown in Fig.1, all anchors are arranged to an anchor map of size $[D, T]$, with x-axis indicating the start time and y-axis indicating the duration of each anchor. D is the predefined max duration of anchors. Next, Feature Align Layer uniformly samples L feature vectors for each anchor, and pools them together. BM Layer designed in [2] serves as align layer and outputs a feature map of size $[C', D, T]$, where each anchor is represented by one vector of C' dimension. L is set as 32 and C' is 256, the first 128 dimension for IoU-APR and the second 128 for Offset-APR. IoU-APR and Offset-APR are used to predict the IoU score and offset for each anchor, and the prediction is also shaped as a map M of size $[D, T]$.

Structure of APR is shown in Fig.4(right). APR generates two map sets, each set includes M_{act} of shape $[m, D, T]$ stems from action branches and M_{uni} of $[D, T]$ from universal branch. In IoU-APR, two sets are denoted as Set_{rIoU} (Eq.(3)) and Set_{cIoU} (Eq.(4)). Both sets all indicate anchor's overlap with actions, but are trained by regression loss and classification loss, respectively.

$$Set_{rIoU} = \{M_{rIoU,act}, M_{rIoU,uni}\} \subset \{\mathbb{R}^{m\times D\times T}, \mathbb{R}^{D\times T}\} \quad (3)$$

$$Set_{cIoU} = \{M_{cIoU,act}, M_{cIoU,uni}\} \subset \{\mathbb{R}^{m\times D\times T}, \mathbb{R}^{D\times T}\} \quad (4)$$

In Offset-APR, two sets are denoted as Set_{cent} (Eq.(5)) and Set_{dura} (Eq.(6)). These two sets indicate each anchor's center offset and duration offset respectively. Sigmoid layer is removed in offset-APR to predict offsets.

$$Set_{cent} = \{M_{cent,act}, M_{cent,uni}\} \subset \{\mathbb{R}^{m\times D\times T}, \mathbb{R}^{D\times T}\} \quad (5)$$

$$Set_{dura} = \{M_{dura,act}, M_{dura,uni}\} \subset \{\mathbb{R}^{m\times D\times T}, \mathbb{R}^{D\times T}\} \quad (6)$$

Similar to BPR, every action branch takes charge of a certain class and the universal branch distills universal patterns, and they are fused by the Filter&Fusion Module as shown in Eq.(7). For concision, M refers to any of $M_{rIoU}, M_{cIoU}, M_{cent}$ and M_{dura}. The weight $\omega_{APR} \in \mathbb{R}^m$ is also trainable ($\mathbf{0} \leq \omega_{APR} \leq \mathbf{1}$).

$$M_{ff} = (\theta \odot \omega_{APR}) \cdot M_{act} + (1 - \theta \cdot \omega_{APR}) M_{uni} \quad (7)$$

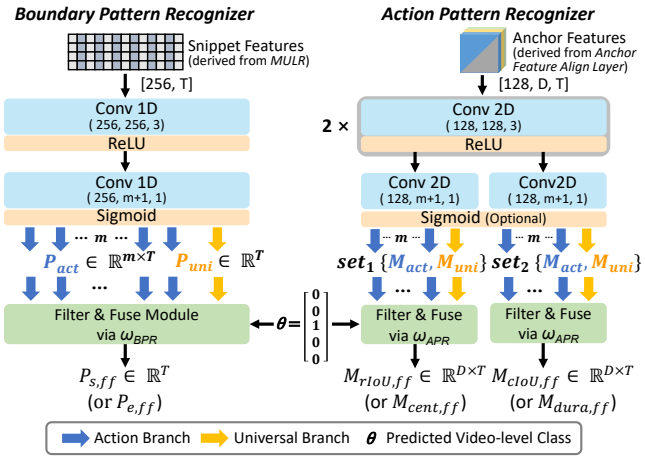


Fig. 4: Boundary Pattern Recognizer (left) and Action Pattern Recognizer (right). In convolution blocks, numbers in brackets represent the input channel, output channel and kernel size respectively.

Finally, we can obtain $P_{s,ff}$,$P_{e,ff}$ of shape $[T]$ from Start-BPR and End-BPR, indicating the boundary probabilities of each snippet. Obtain $M_{rIoU,ff}$ and $M_{cIoU,ff}$ of shape $[D, T]$ from IoU-APR, indicating each anchor's IoU score. Obtain $M_{cent,ff}$ and $M_{dura,ff}$ of shape $[D, T]$ from Offset-APR, indicating each anchor's center offset and duration offset.

2.4. Multi-relations Builder

Multi-relations Builder (MULR) is to fully explore temporal and semantic relations in video simultaneously. As shown in Fig.3, there are three well-designed paths in this module. (1) *Local Path* consists only of 1D convolutions and extract local patterns around certain snippet. (2) In the *Segmental Path*, GRU is introduced to capture segmental temporal relations. Video feature is sent in temporal order to the stacked Fore-GRUs to accumulate valuable information from past moments. Furthermore, considering the whole video feature is accessible in TAD task, feature flipped along time dimen-

sion is sent into stacked Back-GRUs to "memorize" useful information from the future. With the help of GRU, receptive field of model extends bidirectionally to the past and future, which builds the segmental temporal relations. (3) To grasp global semantic relations, two cascaded self-attention modules construct the *Semantic Path*. Inspired by [8], multi-head attention is applied and the head count is set as 4. Attention mechanism builds global relations between any segments with similar semantic information, exchanging and aggregating information as needed. In summary, three basic units (Conv1D, GRU and Attention) are adopted flexibly to explore both temporal and semantic relations. Experiments have verified that MULR can improve performance and reduce computational cost by 38% than graph model GTAD [4], and can also boost some other existing methods like BMN [2] and TSI [3]. Detailed structure of MULR are listed in **Supplementary Materials**.

2.5. Training

Label Assignment. Assuming the video duration is T_v, for the feature of i-th snippet $f_i \in \mathbb{R}^C$ in feature sequence $F_v \in \mathbb{R}^{C\times T}$, its corresponding period is $[(i-1)\cdot\frac{T_v}{T}, i\cdot\frac{T_v}{T}]$. In annotation, a ground-truth action is denoted as $\varphi_g = [t_s^*, t_e^*]$. Expanding boundary from moment to region, the start region is defined as $r_s = [t_s^* - 1.5\cdot\frac{T_v}{T}, t_s^* + 1.5\cdot\frac{T_v}{T}]$ and end region as $r_e = [t_e^* - 1.5\cdot\frac{T_v}{T}, t_e^* + 1.5\cdot\frac{T_v}{T}]$. For BPR, label of start probabilities $G_s \in \mathbb{R}^T$ is obtained by computing the overlap of each snippet with r_s. Label of end probabilities G_e can be obtained in the same way. For APR, calculate IoU between each anchor and all actions, and then arrange the result into a map $G_{IoU} \in \mathbb{R}^{D\times T}$, according to the layout shown in Fig.1. For each anchor $[t_s, t_e]$ whose IoU score larger than 0.9, assuming its corresponding ground truth segment is $[t_s^*, t_e^*]$, calculate the center offset Δc and duration offset Δd as Eq.(8)(9), and arrange all anchors' offsets into maps G_{cent} and G_{dura}.

$$c = \frac{t_s + t_e}{2},\ d = t_e - t_s,\ c^* = \frac{t_s^* + t_e^*}{2},\ d^* = t_e^* - t_s^* \tag{8}$$

$$\Delta c = \frac{c^* - c}{d},\ \Delta d = \log\frac{d^*}{d} \tag{9}$$

Basic Loss. (1) Start-BPR and End-BPR are trained by the re-weighted binary cross-entropy loss L_{bpr}, as shown in Eq.(10). $P = \{p_t\}_{t=1}^T$ is the predicted boundary probabilities and $G = \{g_t\}_{t=1}^T$ is the label. Snippets with $g_t > 0.5$ serve as positive samples (*i.e.*, $\delta\{g_t > 0.5\} = 1$) and others are negative samples. T^+ and T^- represent the number of positive and negative samples respectively. (2) IoU-APR is trained by the re-weighted binary cross-entropy loss L_{apr} (Eq.11) and L2 loss L_2. $M = \{m_i\}_{i=1}^{N_a}$ and $G = \{g_i\}_{i=1}^{N_a}$ represent the predicted value and ground-truth value of each anchor. N_a, N_a^+, N_a^- represent the number of all anchors, positive anchors whose IoU greater than 0.9 and negative anchors, respectively. (3) Offset-APR is trained by the Smooth L1 loss L_1. Note that only those anchors with IoU greater than 0.9 participate in the training of Offset-APR. The re-weights in L_{bpr} and L_{apr} are used to balance the number between positive and negative samples.

$$L_{bpr}(P,G) = -\frac{1}{T}\sum_{t=1}^{T}(\frac{T}{T^+}\cdot\delta\{g_t > 0.5\}\cdot\log p_t + \frac{T}{T^-}(1-\delta\{g_t > 0.5\})\cdot\log(1-p_t)) \tag{10}$$

$$L_{apr}(M,G) = -\frac{1}{N_a}\sum_{i=1}^{N_a}(\frac{N_a}{N_a^+}\cdot\delta\{g_i > 0.9\}\cdot\log m_i + \frac{N_a}{N_a^-}(1-\delta\{g_i > 0.9\})\cdot\log(1-m_i)) \tag{11}$$

Training Objective. Training objective of MrCAN consists of three parts, as shown in Eq.(12)(13)(14), respectively. The total cost is formulated as Eq.(15). λ_1 and λ_2 are balance coefficients between different terms, which are set as 5 and 25 respectively according to the ablation study in Supplementary Materials.

$$L_{bound} = L_{bpr}(P_{s,ff}, G_s) + L_{bpr}(P_{e,ff}, G_e) \tag{12}$$

$$L_{IoU} = L_{apr}(M_{cIoU,ff}, G_{IoU}) + \lambda_1\cdot L_2(M_{rIoU,ff}, G_{IoU}) \tag{13}$$

$$L_{off} = L_1(M_{cent,ff}, G_{cent}) + L_1(M_{dura,ff}, G_{dura}) \tag{14}$$

$$L = L_{bound} + L_{IoU} + \lambda_2\cdot L_{off} \tag{15}$$

2.6. Inference

After obtaining $P_{s,ff}$, $P_{e,ff}$, $M_{rIoU,ff}$, $M_{cIoU,ff}$, $M_{cent,ff}$ and $M_{dura,ff}$ from CAM, following [1, 2, 3], snippet in boundary probability P is screened out as candidate boundary point if it is local peak or its probability is greater than $0.5\cdot max(P)$. All start snippets and end snippets are matched to generate proposals. Proposal starts at t_1 and ends at t_2 is denoted as (t_1, t_2), its center is $c = (t_1+t_2)/2$ and duration is $d = t_2 - t_1$. We can get its start probability $P_{s,ff}^{t_1}$, end probability $P_{e,ff}^{t_2}$ and IoU score $M_{IoU,ff}^{(t_1,d)} = M_{rIoU,ff}^{(t_1,d)}\cdot M_{cIoU,ff}^{(t_1,d)}$. Subsequently, to refine the boundary, we can obtain center offset $\Delta c = M_{cent,ff}^{(t_1,d)}$ and duration offset $\Delta d = M_{dura,ff}^{(t_1,d)}$, and adjust the boundary to (t_1', t_2') as Eq.(16):

$$c' = d\cdot\Delta c + c,\ d' = d\cdot e^{\Delta d}$$
$$t_1' = c' - \frac{d'}{2},\ t_2' = c' + \frac{d'}{2} \tag{16}$$

Finally, proposal-level classifier assigns every proposal with a fine label and a classification score $P_{label}^{(t_1,t_2)}$. The final confidence score $score^{(t_1',t_2')}$ is shown as Eq.(17). Then Soft-NMS [17] is adopted to remove redundant segments.

$$score^{(t_1',t_2')} = P_{s,ff}^{t_1}\ \cdot\ P_{e,ff}^{t_2}\ \cdot\ M_{IoU,ff}^{(t_1,d)}\cdot\ P_{label}^{(t_1,t_2)} \tag{17}$$

3. EXPERIMENTS

3.1. Dataset and Settings

Dataset. We validate the effectiveness of MrCAN on two challenging benchmarks. **ActivityNet-1.3** [18] contains 19994 temporally annotated videos with 200 action classes. Besides, the hierarchical structure of all classes is accessible in annotation. **THUMOS-14** consists of 200 temporally annotated videos in validation set (about 3000 action instances included) and 213 videos in testing set (about 3400 action instances included) . 20 action categories are included.

Implementation Details. As described in Sec.2.2, two-stream network [9] with architecture [10] extracts features of 400 dimension from raw videos at a regular frame interval σ (σ is set as 16 and 5 for ActivityNet and THUMOS, respectively). Meanwhile, features provided by [4, 11, 12] are also used for fair and comprehensive comparison. Following [1, 2, 3], for ActivityNet, we resize feature sequence of all videos to $T = 100$ using linear interpolation. On THUMOS, sliding windows with length of 128 and overlap ratio of 0.5 are applied to trim videos. The max duration D of anchors is set as 100 for ActivityNet and 64 for THUMOS. For fair comparison, following [1, 2, 3, 13, 14], Untrimmed Net [15] serves as classifiers for THUMOS, and the recognition model by [16] for ActivityNet. In annotation of ActivityNet, the hierarchical structure of classes has four layers, including 5/14/53/200 categories respectively. According to ablation studies in Sec.3.4, 14 categories are selected as video-level action classes and the bottom 200 categories as proposal-level

Table 1: Temporal action detection performance comparison with state-of-the-art methods on validation set of ActivityNet-1.3. Bold data indicates the best performance. MrCAN performs better than all current methods, especially the mAP@0.95 at high IoU threshold.

Model	Feat	mAP@0.5	mAP@0.75	mAP@0.95	Avg. mAP
BSN [1]	TS	46.45	29.96	8.02	30.03
BMN [2]		50.07	34.78	8.29	33.85
BC-GNN [7]		50.56	34.75	9.37	34.26
GTAD [4]		50.36	34.60	9.02	34.09
TSI [3]		51.18	35.02	6.59	34.15
BSN++ [21]		51.27	35.70	8.33	34.88
PCMNet [22]		51.35	36.10	9.49	35.27
TCA-Net [13]		52.27	**36.73**	6.86	35.52
MrCAN (ours)		**52.38**	36.51	**10.02**	**35.85**
BUTAL [20]	I3D	43.47	33.91	9.21	30.12
PGCN [5]		48.26	33.16	3.27	31.11
Anchor-free [23]		52.40	35.30	6.50	34.40
RTD-Net [14]		47.21	30.68	8.61	30.83
MrCAN (ours)		**52.90**	**37.32**	**10.53**	**36.53**
GTAD [4]	TSP	51.26	37.12	9.29	35.81
BMN [2]		51.23	36.78	9.50	35.67
MrCAN (ours)		**54.08**	**37.98**	**10.92**	**37.30**

classes. As for THUMOS, due to the lack of hierarchical information, we divide proposal-level 20 actions into 10 video-level classes based on characteristics of actions. **Supplementary Materials** provide details of feature extractor and action classification. MrCAN is trained by Adam optimizer with batch size of 16 and learning rate of 10^{-3}. Training epoch is set as 7 and 10 for THUMOS and ActivityNet respectively. Learning rate is decayed to 10^{-4} after 5 epochs on THUMOS and 7 epochs on ActivityNet.

3.2. Comparison with State-of-the-Art Methods

We compare the proposed MrCAN with state-of-the-art methods on THUMOS-14 and ActivityNet-1.3. On testing set of THUMOS-14, MrCAN reports mAPs of IoU thresholds $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ and the average value of these metrics (*i.e.*,Avg.mAP). On validation set of ActivityNet-1.3, mAPs at IoU $\{0.5, 0.75, 0.95\}$ are reported and Avg.mAP is calculated of IoU thresholds between 0.5 and 0.95 with step of 0.05. For fair and complete comparison, besides the two-stream feature **'TS'**, we also test MrCAN using **'I3D'** [19] features provided by [12] (also used by BUTAL [20]), two-stream features of 2048 dimensions **'TS*'** by GTAD [4] for THUMOS, and **'TSP'** features by TSP [11] for ActivityNet.

On ActivityNet, Table1 reveals that MrCAN outperforms all other existing methods and reaches outstanding performance under three different features. In terms of Avg.mAP, MrCAN improves Avg.mAP to 35.85%, 36.53% and 37.30% using TS, I3D and TSP features, respectively. **Especially at high IoU threshold, MrCAN is the first model whose mAP@0.95 exceeds 10%,** which justifies that predictions of MrCAN are more reliable.

On THUMOS, Table2 shows the superiority of MrCAN. MrCAN greatly outperforms all other methods no matter which feature is used. For instance, using TS feature, MrCAN improves the Avg.mAP from current best 45.78% to 48.98%. As mentioned earlier, different calculation method of Avg.mAP leads to the value difference on two datasets (35% v.s. 48%). However, in terms of mAP@0.5, performances of MrCAN on two datasets are consistent (about 50%, TS features). In addition, despite reaching state-of-the-art performance on both datasets, the improvement of MrCAN on ActivityNet is not so obvious compared with THUMOS. Explanations of this gap are discussed in **Supplementary Materials**.

3.3. Effectiveness of Multi-relations Builder

Multi-relations Builder (MULR) is designed to explore temporal and semantic relations in an unified module effectively. To fully show the edge of MULR, we remove all action branches and compare MrCAN with GTAD [4] which uses graph convolution to model relations. Using TS* feature (provided by GTAD), mAP@0.5 and Avg.mAP on THUMOS are reported in Table3. Computational cost is measured by Multiply–Accumulate Operations (MACs) and calculated using the same input size of $[400, 100]$. Impressively, MrCAN greatly outperforms GTAD with 38% less computational cost.

Furthermore, to justify the generalization of MULR, we choose BMN [2] and TSI [3] which lack explicit construction of relations as baselines, and replace their first three convolution layers with MULR. Experiments are carried out on THUMOS with TS feature. Table3 demonstrates that MULR significantly boosts the performance of baselines, with improvement more than 4% on Avg.mAP. Note that MULR only brings a slight addition of computational cost.

Table 2: Temporal action detection performance comparison with state-of-the-art methods on testing set of THUMOS-14. '†' indicates the reproduced results. MrCAN outperforms other existing methods.

Model	Feat	mAP@0.3	mAP@0.4	mAP@0.5	mAP@0.6	mAP@0.7	Avg. mAP
BSN [1]	TS	53.50	45.00	36.90	28.40	20.00	36.76
BMN [2]		56.00	47.40	38.80	29.70	20.50	38.48
DBG [6]		57.80	49.40	39.80	30.20	21.70	39.78
BC-GNN [7]		57.10	49.10	40.40	31.20	23.10	40.18
BSN++ [21]		59.90	49.50	41.30	31.90	22.80	41.08
TSI [3]		61.00	52.10	42.60	33.20	22.40	42.26
TCA-Net [13]		60.60	53.20	44.60	36.80	26.70	44.38
PCMNet [22]		61.50	55.40	47.20	37.50	27.30	45.78
MrCAN (ours)		**65.54**	**58.60**	**50.22**	**41.00**	**30.56**	**48.98**
GTAD [4]	TS*	66.40	60.40	51.60	37.60	22.90	47.78
BMN† [2]		64.16	59.14	51.97	41.14	30.17	49.32
MrCAN (ours)		**69.65**	**64.16**	**56.17**	**47.12**	**36.15**	**54.65**
PGCN [5]	I3D	63.60	57.80	49.10	-	-	-
BUTAL [20]		53.90	50.70	45.40	38.00	28.50	41.20
RTD-Net [14]		68.30	62.30	51.90	38.80	23.70	49.00
ContextLoc[24]		68.30	63.80	54.30	41.80	26.20	50.88
Anchor-free [23]		67.30	62.40	55.50	43.70	31.10	52.00
MrCAN (ours)		**71.33**	**66.83**	**60.56**	**50.12**	**37.88**	**57.34**

Table 3: Effectiveness of MULR. mAP@0.5 and Avg.mAP on THUMOS are reported. MULR achieves better performance with less computational cost than GTAD. Besides, MULR can also be generalized to BMN and TSI to boost their performances obviously.

Model	Feat	MACs(G)	mAP@0.5	Avg.mAP
GTAD	TS*	45.71	51.60	47.78
MrCAN (w/o Act. Branch)		28.68	**55.68**	**53.98**
BMN	TS	24.62	38.80	38.48
BMN+MULR		25.07	44.19 **(+5.39)**	43.14 **(+4.66)**
TSI		24.68	42.60	42.26
TSI+MULR		25.13	47.21 **(+4.61)**	46.46 **(+4.20)**

3.4. Ablation Study

To substantiate effectiveness of each module and explore influences of different settings, sufficient ablation experiments are conducted.

Ablation study on CAM. The introduction of CAM brings three questions: (1) Does the model parameters and computational cost increase m times after adding m action branches? (2) How should the number of action branches be set? (3) What role do action branches and the universal branch play?

To answer these questions, ablation study on CAM is performed on both datasets. Results on ActivityNet are listed as Table4, and results on THUMOS can be found in Supplementary Materials. (1) For the first question, when increasing action branches from 5 to 200, increase of parameters and computational cost is slight and acceptable. Note that even the computational cost of MrCAN with 200 branches is still 35% less than GTAD. The reason is that action branch is realized by only changing output channels of the last convolution layer. (2) For the second question, as shown in Table4, performance may NOT be better when increasing action branches. A possible explanation is that there is a trade-off between action class number and detection accuracy. Too few action branches causes larger intra-class variance inside each branch and too many branches

reduce the number of effective training samples per branch, which both undermine the performance. 14 branches is reasonable for ActivityNet due to that video-level classifier is responsible for general and sketch classification, and thus the number of action branches should not be too large. (3) For the third question, Table4 reveals that the design of action branch can improve performance compared with model with only universal branch. Moreover, performance can be further boosted if add the universal branch to CAM, which certifies the importance of universal branch.

Table 4: Ablation study on CAM, conducted on ActivityNet using TS feature, mAP@0.95 and Avg. mAP are reported.

Model	#Action Branch	#Universal Branch	Params (M)	MACs (G)	mAP@0.95	Avg. mAP
GTAD	0	1	9.49	45.71	9.02	34.09
MrCAN	0	1	7.60	28.68	9.91	35.24
	5	0	7.61	28.70	9.85	35.46
	5	1	7.61	28.70	9.99	35.57
	14	0	7.62	28.74	10.06	35.53
	14	1	7.62	28.75	10.02	**35.85**
	53	0	7.66	28.95	9.63	35.23
	53	1	7.66	28.95	10.01	35.74
	200	0	7.81	29.72	9.82	35.40
	200	1	7.81	29.72	**10.17**	35.56

Ablation study on MULR. MULR has three paths for local temporal relations, segmental temporal relations and global semantic relations, respectively. Ablation study on different path settings is performed on both datasets with TS feature. Results are listed in Table5. Local path only consists of 1D convolution is a fundamental path. On this basis, adding segmental temporal relations by GRU or adding global semantic relations by self-attention can improve performance, as declared by the first three rows in table. In addition, Complete MULR at the last row verifies that these three paths do not conflict, and combining them is the best choice.

Table 5: Ablation study on the settings of MULR.

Local (Conv1D)	Segmental (GRU)	Semantic (Attention)	THUMOS		ActivityNet	
			mAP@0.5	Avg. mAP	mAP@0.5	Avg.mAP
✓			46.85	45.90	51.85	35.30
✓	✓		49.50 (+2.65)	48.36 (+2.46)	52.24 (+0.39)	35.63 (+0.33)
✓		✓	48.91 (+2.06)	47.39 (+1.49)	52.04 (+0.19)	35.53 (+0.23)
✓	✓	✓	**50.22** (+3.37)	**48.98** (+3.08)	**52.38** (+0.53)	**35.85** (+0.55)

Ablation study on total structure. Remove CAM or MULR from complete MrCAN and test them on both datasets using TS feature. As exhibited in Table6, both CAM and MULR are central to the detection accuracy of MrCAN.

Table 6: Ablation study on total structure.

CAM	MULR	THUMOS		ActivityNet	
		mAP@0.5	Avg. mAP	mAP@0.5	Avg.mAP
✓		46.85	45.90	51.85	35.30
	✓	48.60	46.87	52.01	35.24
✓	✓	50.22	48.98	52.38	35.85

4. CONCLUSION

In this paper, we point out two key aspects in temporal action detection task: recognizing action patterns and exploring multiple relations. We contend that classes with huge differences should be recognized differently, and thus Class-Aware Mechanism with several action branches is proposed. Besides, Multi-relations Builder is designed to explore temporal and semantic relations simultaneously. This builder is composed of three basic units and can also be generalized to some other methods. These two innovations are integrated as Class-Aware Network with Multi-relations (MrCAN). Comprehensive experiments verifies that MrCAN remarkably outperforms existing methods and achieves state-of-the-art performance on both THUMOS-14 and ActivityNet-1.3.

5. REFERENCES

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

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

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

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

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

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

[7] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu, "Boundary content graph neural network for temporal action proposal generation," in *ECCV*, 2020.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[10] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang, "Cuhk & ethz & siat submission to activitynet challenge 2016," *CVPR ActivityNet Workshop*, 2016.

[11] Humam Alwassel, Silvio Giancola, and Bernard Ghanem, "Tsp: Temporally-sensitive pretraining of video encoders for localization tasks," in *ICCV*, 2021.

[12] Daochang Liu, Tingting Jiang, and Yizhou Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *CVPR*, 2019.

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

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

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

[16] Yue Zhao, Bowen Zhang, Zhirong Wu, Shuo Yang, Lei Zhou, Sijie Yan, Limin Wang, Yuanjun Xiong, D Lin, Y Qiao, et al., "Cuhk & ethz & siat submission to activitynet challenge 2017," *arXiv preprint arXiv:1710.08011*, 2017.

[17] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis, "Soft-nms–improving object detection with one line of code," in *ICCV*, 2017.

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

[19] Joao Carreira and Andrew Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[20] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian, "Bottom-up temporal action localization with mutual regularization," in *ECCV*, 2020.

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

[22] Xin Qin, Hanbin Zhao, Guangchen Lin, Hao Zeng, Songcen Xu, and Xi Li, "Pcmnet: Position-sensitive context modeling network for temporal action localization," *arXiv preprint arXiv:2103.05270*, 2021.

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

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