Multi-level Content-Aware and Contextual Perception Network for Temporal Action Proposal Generation

LIU Rui

***Abstract*—Temporal action proposal generation is a crucial component in video understanding, aiming to identify potential action segments from untrimmed videos. However, existing methods often struggle with precise boundary localization and effective modeling of long-range temporal dependencies. To address these challenges, we propose MC-CPN (Multi-level Content-Aware and Contextual Perception Network), a novel framework that integrates multi-layer content representations with both local and global contextual modeling. MC-CPN introduces a Temporal Context-Aware Module (TCAM) to enhance frame-level feature perception and capture long-term dependencies, while a hierarchical fusion strategy bridges frame-level and proposal-level cues for more accurate boundary prediction and confidence estimation. Extensive experiments conducted on Thumos-14 and ActivityNet-1.3 demonstrate that our method achieves superior performance across multiple evaluation metrics, showcasing strong robustness and generalization in diverse temporal action scenarios.**

***Index Terms*—Temporal Action Proposal Generation; Contextual Modeling; Self-Attention Mechanism**

I. INTRODUCTION

Temporal action proposal generation (TAPG) aims to localize potential action segments within untrimmed videos and provide high-quality candidate regions for subsequent temporal action detection. This task is particularly challenging due to the inherently ambiguous nature of action boundaries in continuous video streams. In recent years, boundary-based methods [1-3] have achieved remarkable progress in TAPG by evaluating whether each temporal position can serve as a plausible starting or ending point of an action instance.

Boundary Sensitive Network (BSN) [1] and Boundary-Matching Network (BMN) [2] typically adopt a dual-branch architecture, with one branch responsible for locating action boundaries and the other for evaluating confidence scores. Although these methods improve proposal precision to some extent, a key limitation lies in the decoupled modeling of boundary and content information, which fails to capture the intrinsic interdependencies between them. In practice, the temporal semantic features of action content are closely tied to their corresponding boundary positions, and boundary determination often relies on an accurate understanding of contextual action cues. As a result, methods that lack joint modeling mechanisms tend to produce unstable predictions under scenarios with ambiguous boundaries or drastic semantic changes, severely affecting overall performance. Moreover, existing approaches mainly focus on modeling within local temporal windows, without fully exploring long-range dependencies. This limitation becomes particularly problematic when handling long-duration or semantically extensive actions, often leading to mis-segmentation or inaccurate boundary positioning. Although recent models such as DBG[4] and MSA-Net[5] attempt to enhance temporal representation through multi-scale feature fusion, they still suffer from limited single-level representations and lack effective collaboration across temporal granularities, resulting in suboptimal boundary prediction performance in scenarios with complex variations in action durations.

To address the aforementioned limitations, MCBD[6] introduces a multi-level content-aware boundary detection framework, which formulates multi-scale information modeling as a series of probabilistic prediction tasks. Specifically, it generates boundary and content-related probabilities at both the frame and proposal levels, and further integrates these multi-level representations to effectively capture temporal dependencies. This approach provides a novel perspective and solution for TAPG and has demonstrated strong performance on several public benchmarks. However, MCBD still suffers from limited capacity in modeling long-range dependencies. Its frame-level features rely primarily on local convolutional operations, whose restricted receptive field hampers the ability to represent long-duration and complex actions. Moreover, the simplistic transformation from frame-level to proposal-level features may lead to inadequate temporal modeling when dealing with diverse action patterns.

To tackle the aforementioned issues, this paper draws inspiration from MCBD and focuses on modeling long-range temporal dependencies and complex contextual information from a multi-level content-aware perspective, aiming to improve boundary localization and confidence estimation in temporal action proposals. Based on this, we propose a novel Multi-level Content-aware and Contextual Perception Network (MC-CPN), which comprises three main contributions:

(1) The proposed method effectively integrates complementary information from both action boundaries and semantic content, alleviating the impact of boundary ambiguity and enhancing the stability and robustness of generated proposals;

Manuscript received April 26, 2025
Liu Rui, School of Software, Tiangong University, Tianjin, China

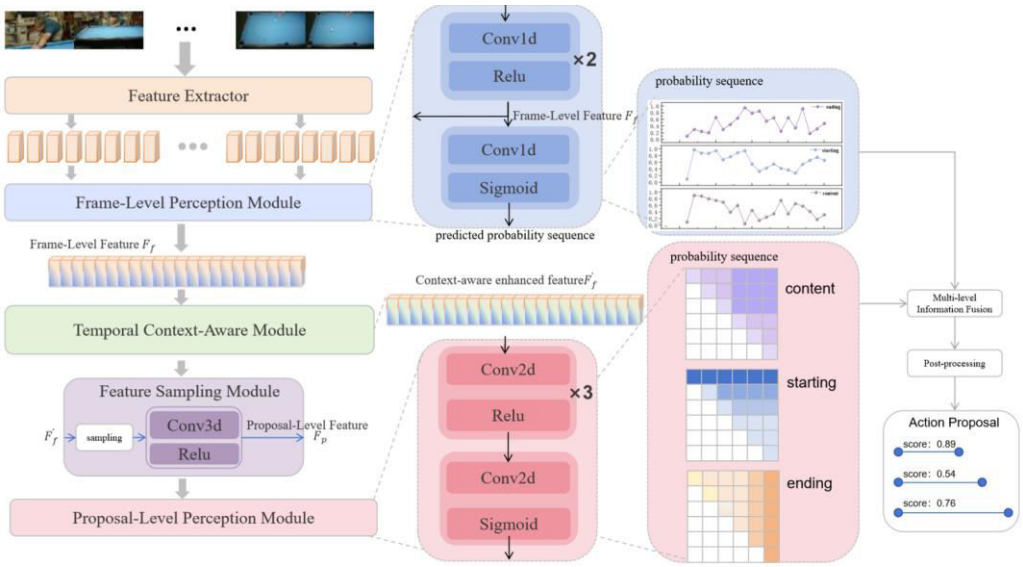

Fig. 1 The overall architecture of the MC-CPN method

(2) A Temporal Context-Aware Module (TCAM) is designed to strengthen the temporal modeling capability of frame-level features via self-attention mechanisms, enabling the model to capture long-duration and structurally complex actions more accurately;

(3) The joint modeling of frame-level and proposal-level features facilitates the multi-level exploitation of temporal cues, leading to improved boundary precision and more reliable confidence estimation, thereby enhancing the overall performance of the system.

II.METHOD

We proposes a Multi-level Content-aware and Contextual Perception Network (MC-CPN) for temporal action proposal generation. The objective of this model is to predict a set of action proposals involving human activities, denoted as $\Phi' = \{(t'^{s}_{n}, t'^{e}_{n}, q'_{n})\}_{n=1}^{N}$, where t'^{s}_{n} and t'^{e}_{n} represent the predicted start and end times of the n-th action instance, and q'_{n} denotes the confidence score of the proposal.

The overall structure of the MC-CPN is shown in Fig. 1. The model primarily consists of four main components: (1) Frame-Level Perception Module: responsible for extracting frame-level features from the input video and generating prediction probability sequences to establish the perception of local temporal information. (2) Temporal Context-Aware Module: models long-term dependencies and optimizes temporal features using self-attention mechanisms to enhance feature representation. (3) Feature Sampling Module: samples one-dimensional features and transforms them into two-dimensional features to model proposal-level features. (4) Proposal-Level Perception Module: generates predicted probability confidence maps based on proposal-level features. The details of each module are elaborated below.

In this study, we use pre-extracted video features as model input and select state-of-the-art feature extractors tailored to each dataset to ensure compatibility with data characteristics and task requirements. Specifically, I3D[7] is adopted for Thumos14, while TSP[8] is used for ActivityNet-1.3.

A. Frame-Level Perception Module

This module takes as input the pre-extracted temporal feature sequence $F = \{f_t\}_{t=1}^{T} \epsilon R^{T \times C}$, where T denotes the length of the feature sequence and C represents the feature dimension. It outputs the frame-level feature representation $F_f \epsilon R^{D_f \times T}$, and estimates the start, end, and content probabilities for each temporal location. Specifically, this module first employs two one-dimensional convolutional layers with ReLU activation to extract local temporal information. The operation is defined as follows:

$$F_f = Relu(Conv1d(Relu(Conv1d(F)))) \quad (1)$$

Then, a one-dimensional convolutional layer with a sigmoid activation function is used as a classifier to obtain the start, end, and content probability sequences for each temporal position, as shown in Equation (2). The content probability indicates the likelihood that a given video segment is part of an action instance.

$$\{X^s, X^e, X^c\} = Sigmoid(Conv1d(F_f)) \quad (2)$$

B. Temporal Context-Aware Module

After extracting the initial temporal features in the Frame-Level Perception Module, the model still faces limitations in capturing long-range dependencies and suffers from insufficient local information. This issue becomes particularly prominent in videos with long-duration actions or complex contextual semantics. To address this challenge, we introduce the Temporal Context-Aware Module (TCAM) to further enhance the modeling of frame-level features. This module effectively integrates global temporal information and rich contextual semantics, thereby strengthening the model's ability to perceive long-range dependencies. As a result, it generates more accurate and semantically expressive feature representations $F'_f \epsilon R^{D'_f \times T}$. The structure of the TCAM module is illustrated in Fig. 2.

To better model the spatial structure of temporal positions, this study draws inspiration from the local neighborhood modeling in Point Transformer[10]. A block extension strategy is used to structurally replicate the initial features F_f obtained from the frame-level perception module along the

feature dimension, extending it to $F_{ext} \epsilon R^{T\times(E\times E)\times D_f}$, where

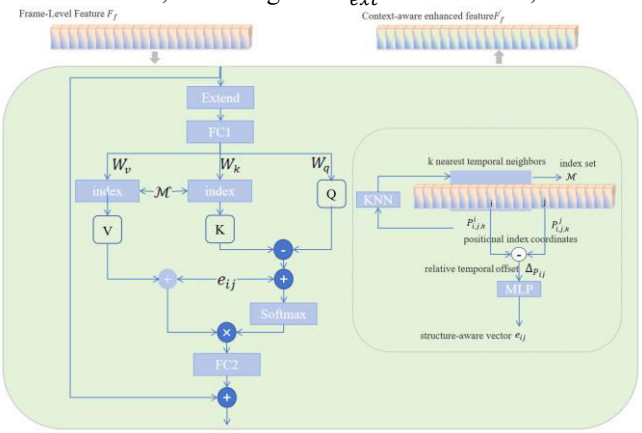

Fig. 2 Architecture of the Temporal Context-Aware Module

E is the extension factor. Simultaneously, for each time step, three-dimensional spatial simulation index coordinates $P_{i,j,k} = (i,j,k), i\epsilon[1,T], j,k\,\epsilon[1,E]$, are constructed. These indexes provide positional references for subsequent feature modeling and neighbor selection. The three-dimensional coordinate information is then flattened so that each position's three-dimensional coordinate information exists as an independent row vector. Finally, the dimensionality is expanded, and the index tensor $P\epsilon R^{(T\times E^2)\times 3}$ is formed, matching the feature tensor in dimensions, with each element associated with a position vector.

Next, to capture the dependencies between time steps, we construct neighborhoods for each time step feature and calculate relative time position encodings. The Euclidean distance squared between the three-dimensional simulation coordinates is used to obtain the distance matrix, as shown in the following equation:

$$dist(i,j) = \left\|P^i - P^j\right\|_2^2 \tag{3}$$

The calculated distance matrix is then sorted, and the k nearest time steps to each time step i are selected as its neighborhood $\mathcal{M}$, where $\mathcal{M}$ represents the index set of the k nearest time steps (the optimal value of k is discussed in detail in the ablation study).

Additionally, to perceive the neighborhood structure, relative time position encoding is introduced. For each pair of time steps i and j, the relative time position difference $\Delta_{P_{ij}}$ is computed as:

$$\Delta_{P_{ij}} = P^i - P^j \tag{4}$$

The calculated relative time position difference is then input into a multilayer perceptron (MLP) for position encoding mapping, yielding the structural awareness vector e_{ij} for each pair of time steps:

$$e_{ij} = \phi\left(\Delta_{P_{ij}}\right) = MLP_\delta(\Delta_{P_{ij}}) \tag{5}$$

Here, e_{ij} is the structural awareness vector obtained after the relative time position difference $\Delta_{P_{ij}}$ is mapped through the MLP, which helps the model capture the relative temporal dependencies between time steps.

Subsequently, the input frame-level extended features are linearly transformed. Through a fully connected mapping layer (i.e., FC_1), the original feature dimension D_f is projected to the uniformly used feature dimension D_{model}:

$$F' = FC_1(F_{ext}) \tag{6}$$

Subsequently, Q, K, and V vectors are constructed. The query vector q_i is generated from the feature of the current time step, representing the direction of the temporal information the current time step "wants to focus on" determining from which part of the temporal data the model should gather information. The key vector k_i and value vector v_i come from the features of adjacent time steps. k_i represents the "reference" information of the features at adjacent time steps, indicating the similarity between the features of the current and adjacent time steps; v_i carries the actual contextual content of the adjacent time step, i.e., it provides specific information to the current time step. The formulas are as follows:

$$Q = W_qF', K = Index(W_kF', \mathcal{M}), V = Index(W_vF', \mathcal{M}) \tag{7}$$

where $W_q, W_k, W_v\,\epsilon R^{D_{model}\times D_{model}}$, $\mathcal{M}$ stores the indices of the k nearest time steps for each time step i. The $Index$ operation extracts the neighbor features for each time step according to the selected neighbor indices, ensuring that the model can adaptively focus on the most relevant temporal features.

Next, we compute the attention scores for each time step and calculate the weights between neighbors based on these scores. The attention score computation formula is:

$$\alpha_{ij} = Softmax(\frac{1}{\sqrt{d}} \cdot MLP_\gamma(q_i - k_i + e_{ij})) \tag{8}$$

where α_{ij} is the attention weight of time step i to time step j, and MLP_γ is the attention score mapping network. Then, using the attention weight, we perform a weighted aggregation of the neighborhood value vectors and position encodings to obtain the enhanced feature representation:

$$z_i = \sum_{j\in\mathcal{M}_i} \alpha_{ij} \cdot (v_i + e_{ij}) \tag{9}$$

Finally, a fully connected mapping layer FC_2 is used to map the aggregated feature z_i back to the original feature space and perform residual connection with the input feature to obtain the final feature representation:

$$f_i' = FC_2(z_i) + f_i \tag{10}$$

The final output is the enhanced frame-level contextual feature $F_f' = \{f_i'\}_{i=1}^T \epsilon R^{D_f'\times T}$, which possesses stronger contextual awareness.

C. Feature Sampling Module

This module converts the previously obtained feature F_f' into a 2D relational matrix to capture the correlation between different time intervals and obtain proposal-level features. First, a sampling matrix A is constructed, where each proposal's start boundary i and end boundary j correspond to a sampling matrix $a_{i,j}$. For the time interval from i to j, N_s uniformly sampled time points $\{t_1, t_2, \dots, t_{N_s}\}$ are generated. For a proposal $\phi_{i,j} = (t_s, t_e)$ ·, a set of evenly spaced time points is defined, as shown in the following formula:

$$t_n = t_s + \frac{n}{N_s}(t_e - t_s), n = 0,1,2,\dots,N_s - 1 \tag{11}$$

For each time point t_n, its floor $\lfloor t_n \rfloor$ and ceiling $\lceil t_n \rceil$ values are used as the frame index values to update the corresponding weight in $a_{i,j}$, as described by the following formula:

$$a_{i,j}[n,t] = \begin{cases} 0.5, & t = \lfloor t_n \rfloor \; or \; \lceil t_n \rceil \\ 0, & otherwise \end{cases} \tag{12}$$

The final output is a vector of length T representing the sampling weights of the time segment from i to j. All $a_{i,j}$ are

combined to form the entire sampling matrix $A \epsilon R^{T \times N_s \times T}$. Then, the feature F_f' is multiplied by the sampling matrix A through a dot product to generate the preliminary proposal-level features $F_p' \epsilon R^{D_f \times N_s \times T \times T}$, as shown in the following formula:

$$F_p' = F_f' \cdot A \tag{13}$$

Since F_p' is a high-dimensional feature representation with a large computational cost, it is downsampled using a 3D convolution to obtain the final proposal-level feature F_p, as shown in the formula:

$$F_p = Conv3d(F_p') \tag{14}$$

D. Proposal-Level Perception Module

This module utilizes the proposal-level features obtained above to generate probability maps. It uses three 2D convolution layers with ReLU activation, followed by a 2D convolution layer with a sigmoid activation to generate the starting, ending, and content probability maps, as follows:

$$\{M^s, M^e, M^c\} = Sigmoid(Conv2d((Relu\left(Conv2d(F_p)\right))^3)) \tag{15}$$

Finally, the starting, ending, and content probability sequences corresponding to each time position obtained from the frame-level perception module are integrated with the starting, ending, and content probability maps generated by the proposal-level perception module through the multi-level information fusion mechanism. This process results in the final temporal action proposals, which are then scored. Afterward, the proposals undergo a Soft-NMS[9] post-processing operation to eliminate redundant action proposals, ensuring that the outputted action proposals are more precise and reliable.

E. Training Methodology

To supervise the learning of probability sequences and probability maps, we construct the ground truth label sequence $\{S^s, S^e, S^c\} = \{S_t^s, S_t^e, S_t^c\}_{t=1}^T$ and the ground truth label map $\{H^s, H^e, H^c\} = \{h_{i,j}^s, h_{i,j}^e, h_{i,j}^c\}_{i=1,j=2}^{i<j\leq T}$, as described in[1]. The value of 1 in S^s or S^e at time step t indicates the presence of a start or end boundary for an action at time t, while a value of 0 indicates the absence of a start or end boundary. A value of 1 in S^c between time steps t_s and t_e indicates the existence of a real action instance within this time interval. Regarding the ground truth label map, $h_{i,j}^s$ is set to S_i^s, and $h_{i,j}^e$ is set to S_j^e. The intersection-over-union (IoU) of time regions r_1 and r_2 can be defined as $IoU(r_1, r_2) = |r_1 \cap r_2| / |r_1 \cup r_2|$. r_{ij} represents the region between time positions i and j, and $r_{t_s t_e}$ represents the region of the action instance. Then, $h_{i,j}^c = IoU(r_{ij}, r_{t_s t_e})$.

By integrating the generated probability sequences and probability maps, the final loss function consists of both frame-level and proposal-level components:

$$\mathcal{L}_{MC-CPN} = \mathcal{L}_f + \mathcal{L}_p \tag{16}$$

The frame-level loss is defined as:

$$\mathcal{L}_f = \mathcal{L}_{bl}(X^s, S^s) + \mathcal{L}_{bl}(X^e, S^e) + \mathcal{L}_{bl}(X^c, S^c) \tag{17}$$

where $\mathcal{L}_{bl}$ denotes the weighted binary logistic regression loss[11].

The proposal-level loss is defined as:

$$\mathcal{L}_p = \acute{\mathcal{L}}_{bl}(M^s, H^s) + \acute{\mathcal{L}}_{bl}(M^e, H^e) + \acute{\mathcal{L}}_{bl}(M^{cc}, H^c) + \mu \mathcal{L}_{sl}(M^{cr}, H^c) \tag{18}$$

Here, $\mathcal{L}_{sl}$ refers to the smooth L1 loss, and μ is set to 10. $\acute{\mathcal{L}}_{bl}$ represents the two-dimensional version of the weighted binary logistic regression loss.

III. Experiment Setup and Results Analysis

A. Datasets and Evaluation Metrics

Datasets: We evaluates the proposed method on two challenging public datasets: Thumos14 [12] and ActivityNet-1.3 [13].

Thumos14 is widely used for action recognition and temporal action detection tasks. Its training set consists of the UCF101 dataset[14], which includes 13,320 trimmed video clips from 101 action categories. The validation and test sets contain 1,010 and 1,574 untrimmed videos, respectively. Following the standard setup in[1-3], we use 200 temporally annotated videos from the validation set for training, and 213 annotated videos from the test set for evaluation.

ActivityNet-1.3 is a large-scale video dataset designed for action recognition and temporal action localization. It consists of 19,994 videos annotated with 200 action categories. The dataset is split into training, validation, and test sets in a 1:1:2 ratio.

Evaluation Metrics: We use two main evaluation metrics for the temporal action proposal generation task. The first is Average Recall (AR), which is typically calculated as the recall at different numbers of proposals, denoted as AR@AN (Average Recall at a given number of proposals). It measures the proportion of ground-truth actions that are correctly recalled by the model under a specific number of proposals. The second metric is the Area Under the AR vs. AN Curve (AUC), which evaluates the overall recall performance of the model across the entire proposal range. Specifically, for the Thumos14 dataset, AR is measured at various numbers of proposals (AN), including [50, 100, 200, 500, 1000]. For the ActivityNet-1.3 dataset, the evaluation focuses on AR@100 and AUC as key indicators.

B. Temporal Action Proposal Generation Results

Table I presents the performance comparison of our method against several mainstream proposal generation approaches on the Thumos-14 dataset (with the best performance highlighted in bold and the second-best in italics). With I3D features, MC-CPN demonstrates superior performance across multiple evaluation metrics, consistently outperforming existing methods. The integration of the Temporal Context-Aware Module (TCAM) enhances the model's ability to capture long-range temporal dependencies, leading to improved boundary prediction, especially in complex or ambiguous scenarios. Notably, MC-CPN achieves robust results under both low and high proposal settings, indicating strong generalization and reliability in various proposal budgets.

To further evaluate the generalizability of the proposed model and its effectiveness on long-duration videos, we conducted temporal action proposal generation experiments on the ActivityNet-1.3 dataset, using AR@100 and AUC as evaluation metrics. As shown in Table II, MC-CPN achieves the highest AUC of 69.98% under TSP features, outperforming all baseline methods. These results demonstrate that the proposed multi-level content-aware and context modeling strategy is not only effective in short and

diverse scenarios (e.g., Thumos-14), but also exhibits strong robustness and generalization capabilities in long and structurally complex video tasks.

Table I Comparison of Results on Thumos14

Method	Feature	@50	@100	@200	@500	@1000
BSN[1]	TSN	37.46	46.06	53.21	60.64	64.52
BMN[2]	TSN	39.36	47.72	54.70	62.07	65.49
DBG[15]	TSN	37.32	46.67	54.50	62.21	66.40
BSN++[3]	TSN	42.44	49.84	57.61	65.17	66.83
TCANet[16]	TSN	42.05	50.48	57.13	63.61	66.88
AEI-B[17]	C3D	44.97	50.13	57.34	64.43	67.78
AEI-G[17]	C3D	45.31	51.12	58.19	64.58	67.96
ABN[18]	C3D	34.25	44.01	52.05	60.57	65.39
ABN[18]	TSN	40.87	49.09	56.24	63.53	67.29
DCAN[19]	TSN	42.65	51.05	57.95	64.58	68.37
BCNet[20]	TSN	**45.50**	53.60	60.00	67.00	69.80
MCBD[6]	I3D	44.45	53.99	61.17	67.96	70.91
MC-CPN	I3D	*44.91*	**54.41**	**61.63**	**67.98**	**71.10**

Table II Comparison of Results on ActivityNet-1.3

Method	Feature	AR@100	AUC
BSN[1]	TSN	74.16	66.17
BMN[2]	TSN	75.01	67.10
DBG[15]	TSN	76.65	68.23
BSN++[3]	TSN	76.52	68.26
TCANet[16]	TSN	76.08	68.08
AEI-B[17]	C3D	77.25	69.43
AEI-G[17]	C3D	77.24	69.47
ABN[18]	C3D	76.72	69.16
ABN[18]	TSN	76.39	68.84
DCAN[19]	TSN	75.71	67.93
MCBD[6]	TSP	**78.29**	69.90
MC-CPN	TSP	*78.26*	**69.98**

C. Ablation Study Analysis

To comprehensively evaluate the impact of the Temporal Context-Aware Module (TCAM) on proposal generation, we conducted two ablation studies. The first examines the effect of TCAM placement within the MC-CPN framework, while the second investigates the influence of different neighborhood sizes (k) on temporal context modeling.

(1) TCAM Placement Analysis: To identify the optimal insertion point for TCAM, we tested its placement at various stages of the MC-CPN pipeline:

a. No TCAM (baseline);

b. Before the first convolution in the frame-level perception module (early stage);

c. Before the second convolution (middle stage);

d. Before the third convolution (late stage);

e. Before proposal-level feature extraction.

These configurations were compared to assess their effects on temporal action proposal performance.\

Fig. 3 presents the experimental results for different TCAM placement strategies, evaluated using the AR@AN metric on the Thumos-14 dataset. The results indicate that placing TCAM at position e (before proposal-level feature extraction) yields the best performance.

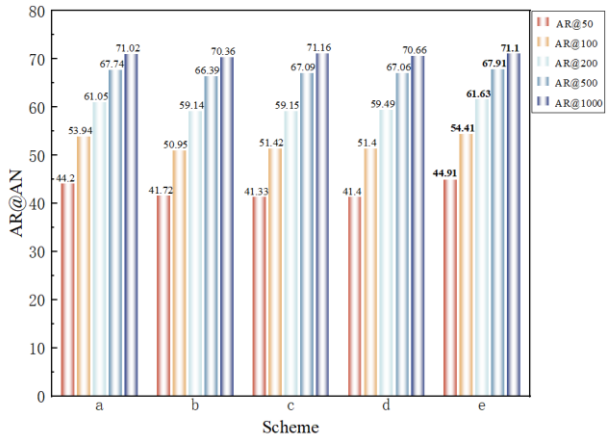

Fig. 3 Experimental Study on TCAM Placement

Specifically, the baseline configuration (position a) without TCAM shows the poorest results, highlighting the model's limited ability to capture temporal context when relying solely on raw features. Inserting TCAM at the early stage of the frame-level perception module (position b) leads to marginal improvements, mainly in AR@50 to AR@100. Position c (middle stage) offers further gains in AR@100 to AR@500, suggesting enhanced context modeling at this depth. Position d (late stage) continues to improve AR@200 and AR@500 performance. Ultimately, position e achieves the best overall results, with significant improvements in AR@500 and AR@1000, confirming that this placement maximizes the benefits of temporal context modeling and substantially enhances proposal accuracy.

(2) Effect of Different k Values

Based on the optimal TCAM placement identified earlier, we further explored the impact of varying neighborhood sizes (k = 2, 4, 6, 8, 16) on model performance. The results, shown in Table III, indicate that k = 4 consistently achieves the best performance.

As k increases, the model captures more temporal dependencies; however, when k reaches 16, excessive contextual information introduces noise and degrades performance. At k = 4, the model strikes an optimal balance between short- and long-range dependencies, leading to the highest detection accuracy. These results demonstrate that moderate k values enhance temporal modeling, while overly large values introduce redundancy that harms boundary localization.

Table II Comparison of Model Performance with Different k Values

	2	4	6	8	16	@50	@100	@200	@500	@1000
k	✓					44.17	54.04	61.36	67.82	71.14
		✓				44.91	54.41	61.63	67.91	71.10
			✓			44.53	54.19	61.11	67.92	71.13
				✓		44.52	54.06	61.20	67.89	71.02
					✓	44.61	54.17	61.30	68.01	71.19

D. Visualization Results Analysis

To provide a more intuitive demonstration of the experimental results, we visualized the action proposals generated by our method on selected videos from the Thumos-14 and ActivityNet-1.3 datasets, as shown in Figures 3-5 and 3-6. In these figures, "GT" denotes the

ground truth action segments.

In Fig. 1, Example A presents a gymnastics action clip with a ground truth interval from 8.5s to 51.2s. Our method, MC-CPN, generated a proposal spanning from 9.0s to 50.4s, which exhibits a higher temporal overlap with the ground truth compared to MCBD. This indicates that MC-CPN is more accurate in capturing complete actions. Example B illustrates a rock climbing action, while Example C shows an athletic throwing action. The results from both examples demonstrate that MC-CPN is capable of generating high-quality proposals regardless of whether the video contains a single action segment or multiple consecutive actions.

presents the visualization results on the ActivityNet-1.3 dataset. As shown in the figure, our method demonstrates superior performance in both samples A and B.

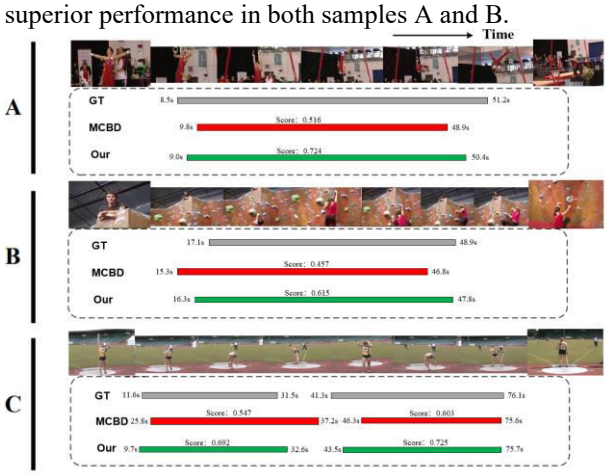

Fig. 4 Proposal Visualization for Thumos-14

Fig. 5 Proposal Visualization for ActivityNet-1.3

V.CONCLUSION

In this paper, we proposed MC-CPN, a temporal action proposal generation method that integrates multi-level content perception and contextual modeling. Extensive experiments on Thumos-14 and ActivityNet-1.3 demonstrate that our method achieves superior performance in both short and long video scenarios. The proposed TCAM effectively enhances long-range dependency modeling, and ablation studies further confirm the benefit of its placement and parameter settings. Overall, MC-CPN shows strong generalization and robustness in complex temporal action tasks.

REFERENCES

[1] Lin T, Zhao X, Su H, et al. Bsn: Boundary sensitive network for temporal action proposal generation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[2] Lin T, Liu X, Li X, et al. Bmn: Boundary-matching network for temporal action proposal generation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3889-3898.

[3] Su H, Gan W, Wu W, et al. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(3): 2602-2610.

[4] Lin C, Li J, Wang Y, et al. Fast learning of temporal action proposal via dense boundary generator[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 11499-11506.

[5] Yang W, Zhang T, Mao Z, et al. Multi-scale structure-aware network for weakly supervised temporal action detection[J]. IEEE transactions on image processing, 2021, 30: 5848-5861.

[6] Su T, Wang H, Wang L. Multi-level content-aware boundary detection for temporal action proposal generation[J]. IEEE Transactions on Image Processing, 2023, 32: 6090-6101.

[7] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset[C]//proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6299-6308.

[8] Alwassel H, Giancola S, Ghanem B. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 3173-3183.

[9] Bodla N, Singh B, Chellappa R, et al. Soft-NMS--improving object detection with one line of code[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5561-5569.

[10] Zhao H, Jiang L, Jia J, et al. Point transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 16259-16268.

[11] Maalouf M, Siddiqi M. Weighted logistic regression for large-scale imbalanced and rare events data[J]. Knowledge-Based Systems, 2014, 59: 142-148.

[12] Idrees H, Zamir A R, Jiang Y G, et al. The thumos challenge on action recognition for videos “in the wild”[J]. Computer Vision and Image Understanding, 2017, 155: 1-23.

[13] Caba Heilbron F, Escorcia V, Ghanem B, et al. Activitynet: A large-scale video benchmark for human activity understanding[C]//Proceedings of the ieee conference on computer vision and pattern recognition. 2015: 961-970.

[14] Soomro K, Zamir A R, Shah M. UCF101: A dataset of 101 human actions classes from videos in the wild[J]. arXiv preprint arXiv:1212.0402, 2012.

[15] Lin C, Li J, Wang Y, et al. Fast learning of temporal action proposal via dense boundary generator[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 11499-11506.

[16] Qing Z, Su H, Gan W, et al. Temporal context aggregation network for temporal action proposal refinement[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 485-494.

[17] Vo K, Joo H, Yamazaki K, et al. Aei: Actors-environment interaction with adaptive attention for temporal action proposals generation[J]. arXiv preprint arXiv:2110.11474, 2021.

[18] Vo K, Yamazaki K, Truong S, et al. Abn: Agent-aware boundary networks for temporal action proposal generation[J]. IEEE Access, 2021, 9: 126431-126445.

[19] Chen G, Zheng Y D, Wang L, et al. Dcan: improving temporal action detection via dual context aggregation[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(1): 248-257.

[20] Yang H, Wu W, Wang L, et al. Temporal action proposal generation with background constraint[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 3054-3062.