Applied Intelligence (2023) 53:15516–15536
https://doi.org/10.1007/s10489-022-04261-1

ORIGINAL SUBMISSION

Exploiting semantic-level affinities with a mask-guided network for temporal action proposal in videos

Yu Yang[1] · Mengmeng Wang[1] · Jianbiao Mei[1] · Yong Liu[1]

Accepted: 10 October 2022 / Published online: 22 November 2022

Abstract

Temporal action proposal (TAP) aims to detect the action instances' starting and ending times in untrimmed videos, which is fundamental and critical for large-scale video analysis and human action understanding. The main challenge of the temporal action proposal lies in modeling representative temporal relations in long untrimmed videos. Existing state-of-the-art methods achieve temporal modeling by building local-level, proposal-level, or global-level temporal dependencies. Local methods lack a wider receptive field, while proposal and global methods lack the focalization of learning action frames and contain background distractions. In this paper, we propose that learning semantic-level affinities can capture more practical information. Specifically, by modeling semantic associations between frames and action units, action segments (foregrounds) can aggregate supportive cues from other co-occurring actions, and nonaction clips (backgrounds) can learn the discriminations between them and action frames. To this end, we propose a novel framework named the Mask-Guided Network (MGNet) to build semantic-level temporal associations for the TAP task. Specifically, we first propose a Foreground Mask Generation (FMG) module to adaptively generate the foreground mask, representing the locations of the action units throughout the video. Second, we design a Mask-Guided Transformer (MGT) by exploiting the foreground mask to guide the self-attention mechanism to focus on and calculate semantic affinities with the foreground frames. Finally, these two modules are jointly explored in a unified framework. MGNet models the *intra-semantic similarities* for foregrounds, extracting supportive action cues for boundary refinement; it also builds the *inter-semantic distances* for backgrounds, providing the semantic gaps to suppress false positives and distractions. Extensive experiments are conducted on two challenging datasets, ActivityNet-1.3 and THUMOS14, and the results demonstrate that our method achieves superior performance.

Keywords Temporal action proposal generation · Temporal action localization · Attention · Transformer

1 Introduction

With the explosive growth of video data on the Internet, video summary tasks have become increasingly crucial for video understanding. Human action understanding tasks are also long-term research goals owing to their various applications. Temporal action localization (TAL), a combination of the above two tasks, has attracted the attention of many researchers. It aims to detect action instances in an untrimmed video by predicting the corresponding starting times, ending times, and action categories. It has various application scenarios, such as intelligent security surveillance, human behavior analysis, and video editing. For instance, TAL can detect abnormal human behaviors to remind people timely in security surveillance. It can also boost human behavior analysis by localizing keyframes. For video editing, it can detect wonderful segments and help extract the highlights of a video. Temporal action localization can be divided into temporal action proposals (TAP) and action recognition. Recently, with the success of action recognition in short-trimmed videos, more attention has been focused on the

✉ Yong Liu
yongliu@iipc.zju.edu.cn

Yu Yang
yu.yang@zju.edu.cn

Mengmeng Wang
mengmengwang@zju.edu.cn

Jianbiao Mei
jianbiaomei@zju.edu.cn

[1] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310027, China

TAP task. It determines where the action events occur in a long untrimmed video, generally implemented by action boundary localization and action proposal scoring. The diverse action contents in untrimmed videos make temporal action proposals a more compelling yet challenging task in video analysis.

The main challenge of temporal action proposals is exploiting temporal relations between different timestamps. Depending on the scale of temporal correlation modeling, previous methods can be categorized into three types: *local-level*, *proposal-level*, and *global-level* methods. Local-level methods utilize ConvNets to regress predefined anchors' boundaries [15–17, 24, 32, 50, 59] or evaluate each frame's actionness probability [25, 34, 47, 56, 57]. However, these local-level methods only exploit inadequate local information and lack a wider receptive field. Proposal-level methods construct proposal representations to capture more temporal contexts. They adopt 2D convolution on the proposal feature map [22, 23] or the Graph Convolutional Networks (GCNs) [2, 48, 54, 55] to model the proposals' relationships. Global-level methods [18, 31, 35, 45, 60, 63] adopt the *query-and-retrieval* procedure or the video transformer to encode global temporal inter-dependencies. However, these proposal-level and global-level methods lack the focalization of learning the semantic similarities/distances between each frame with the action segments.

We argue that semantic-level affinities usually contain more practical information than global relations. As shown in Fig. 1, an action video usually contains many background clips, such as 'cheering', which is easy to be wrongly classified as a false positive sample due to its similar appearances with foregrounds. Therefore, exploiting semantic-level similarities is necessary to determine the distinctions and further boost the action localization. Based on the intuition above, this paper investigates the semantic-level affinities between foregrounds (action segments) and backgrounds (nonaction frames) from two perspectives: 1) *Intra-semantic similarity*: for the foreground segments, learning the intra-semantic similarities from other action clips can extract more supportive cues. For example, in Fig. 1, the table tennis table in the latter frames provides scene information for the former action clip, which lacks practical scenes due to the camera pose. The supportive cues from the latter action clip enhance the former's belief that the action is more likely to be playing table tennis, thus achieving more precise action localization. 2) *Inter-semantic similarity*: capturing the inter-semantic distance between background segments and foregrounds helps discriminate their similar appearances or motion patterns, thus suppressing false positives and background distractions.

In this paper, we focus on exploiting semantic-level affinities to capture more efficient information for the TAP task. To this end, we focus on two problems: *calculating semantic similarity with which segments* and *how to mine semantic associations*. 1) For the first problem, we argue that action-relevant frames are essential to be focused on since they always contain more informative video scenarios. So we set the action frames as the foregrounds, allowing each frame to learn the semantic affinities between them with the actions. 2) For the second question, we design a mask-guided self-attention mechanism. It promotes the foreground segments to aggregate action cues by computing the intra-semantic similarity. It also guides the backgrounds to calculate the inter-semantic distance to restrain false positives.

Specifically, we present a novel framework named MGNet for the temporal action proposal task. MGNet fully

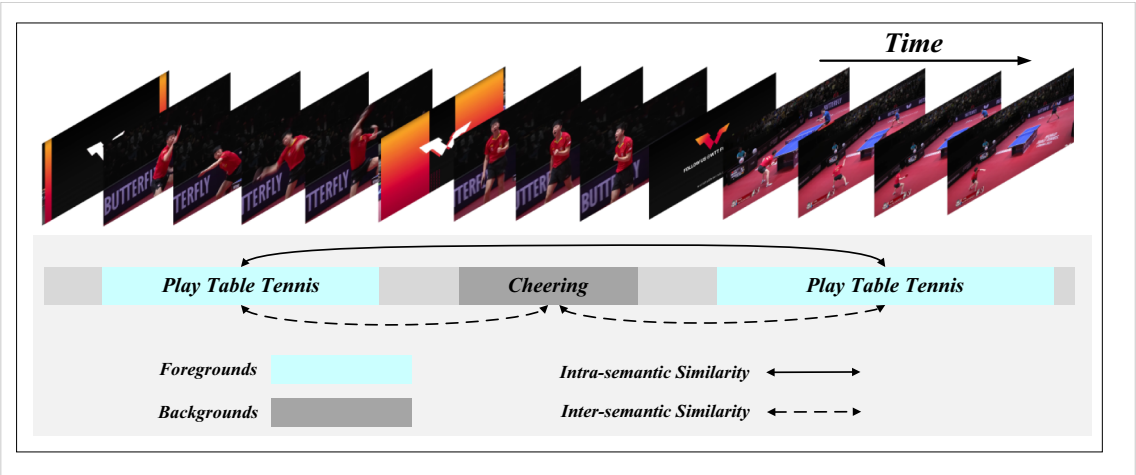

Fig. 1 Illustration of our motivation. A *play table tennis* video contains a *cheering* segment, which is often wrongly classified as a false positive. Learning intra-semantic similarity of foregrounds can extract supportive cues form co-occurring action clips to refine action boundaries. Modeling inter-semantic distance between backgrounds with foregrounds can suppress false positives and distractions

exploits the semantic-level affinities to provide supportive cues for action segments and capture discriminations for background clips. We first propose a Foreground Mask Generation (FMG) module to adaptively generate the foreground mask, representing the locations of the action-relevant frames throughout the video. Second, we design a Mask-Guided Transformer (MGT). It exploits the foreground mask to guide the self-attention mechanism to build semantic associations. In this way, foreground predictions can refine their action boundaries based on the learned action cues; background distractions and false positive predictions are suppressed according to the semantic gaps. Finally, MGNet can be trained end-to-end from scratch to learn semantic-level affinities adaptively.

We evaluate our proposed method on two popular benchmarks, *i.e.*, ActivityNet-1.3 [6] and THUMOS14 [21], for the TAP and TAL tasks. Experimental results demonstrate that our MGNet outperforms the state-of-the-art TAP methods, with the AUC reaching 68.85% on ActivitNet-1.3 and 45.4%@50 on THUMOS14. It also exceeds in the TAL task, with an average mAP that reaches 48.4% on THUMOS14, surpassing RTD-Net [35] 4.8% and TCA-Net [31] 4.0%, respectively.

In summary, our main contributions are as follows:

- We propose a novel MGNet for the TAP task. To the best of our knowledge, this is the first work that exploits semantic-level affinities with foregrounds to capture semantic associations for TAP.
- We design a Foreground Mask Generation (FMG) module to generate the foreground mask representing the timestamps of the action-related frames, which are then employed as prior knowledge to guide the self-attention mechanism learning semantic similarities.
- We propose a Mask-Guided Transformer (MGT), which exploits the foreground mask to model the semantic-level affinities. It models the intra-semantic similarities for foregrounds, providing supportive cues to refine their action boundaries. It also determines the inter-semantic distances for backgrounds, suppressing false positives and distractions.
- Extensive experiments demonstrate that our method outperforms the state-of-the-art methods in temporal action proposal and temporal action localization tasks on ActivityNet-1.3 and THUMOS14 datasets.

2 Related works

2.1 Video understanding

Nowadays, huge amounts of video data are generated in people's daily social contact and security monitoring. As a result, automatic video understanding has become increasingly important and a hot topic with a wide range of applications. For example, for video surveillance, person re-identification (Re-ID) technology [52] can help to retrieve surveillance videos, action recognition technology [20] can monitor dangerous behaviors, small objects detection [30] can detect contraband from security videos. In addition, the knowledge of 2D video understanding can be extended to other vision tasks, such as visual odometry [62], point clouds analysis [19], 3D object detection [44], and depth map estimation [36]. We aim at the video temporal action proposal task, which localizes the human action segments from a long video. It can be applied to intelligent monitoring and extended to video summary and recommendation tasks.

2.2 Video action recognition

Action recognition is a fundamental task in the video understanding domain which can be extended to downstream tasks such as temporal action localization and video captioning, etc. Since action recognition is a foundational and critical task, many algorithms have been proposed, coarsely divided into three types: two-stream networks, 3D-CNNs, and transformer-based networks. Two-steam-based methods [11, 14, 39, 42, 51] exploit RGB flow to capture the spatial information while utilizing optical flow to obtain the temporal features, and then fuse the two types of features to implement action classification. 3D-CNNs [8, 13, 20, 40] extend the common 2D-CNNs with an additional temporal dimension to simultaneously learn spatiotemporal features in the video. Transformer-based networks [1, 3, 29] take full advantage of attention's global scope and employ recent strong Vision Transformer [1] to encode spatial and temporal features jointly. With much exploration, the accuracy of action recognition algorithms is getting more accurate. However, their training data must be trimmed videos that only contain action segments, while actions are always randomly distributed in long videos in reality. Thus, generating actions proposals from untrimmed videos is vital for practical application.

2.3 Temporal action proposal

The objective of TAP is to identify the temporal boundaries of action instances from an untrimmed video, where building temporal correlations are vital for accurate detections. According to the different scales of temporal modeling, current methods can be roughly divided into three categories: local-level, proposal-level, and global-level methods. Local methods can be summarized as *anchor regression* paradigm [15, 17, 24, 32] and *actionness probability* paradigm [9, 23, 25, 34, 57]. The former paradigm employs predefined anchors to regress the action

boundaries, lacking temporal flexibility. While the latter evaluates each frame's boundary probabilities to generate action proposals, which are temporal-flexible. However, they only exploit the local context and lack a wider receptive field. Proposal-level methods [2, 22, 23, 34, 48, 54] tried to introduce proposal-level features by constructing the proposal map or employing GCNs to make up for the deficiency of local information. Global-level methods [31, 35, 63] introduced the self-attention mechanism or the original transformer detection framework (DETR) [7] to build long-range dependency. However, instinctively modeling the global-level temporal dependency lacks the focalization of learning action features and also brings background distractions. On the contrary, our framework exploits the semantic-level affinities with foregrounds to mine temporal associations, improving detection accuracy and suppressing false positives and interferences.

2.4 Transformer and attention mechanism

Transformer was firstly introduced by [37] in the machine translation task. Inspired by the recent advances in NLP tasks, many transformer-based frameworks have also been applied to better suit video understanding tasks like object tracking [49], video instance segmentation [43], and video object segmentation [12]. In addition, to improve the transformer's learning ability and efficiency, some improved self-attention mechanisms were also explored. For example, Big Bird [53] designs a sparse attention mechanism to reduce the computational complexity for long sequences. Wang et al. [41] proposes an aggregate attention module to classify fine-grained images accurately with fewer parameters. VoTr [28] proposes local and dilated attention to enlarge the attention range while maintaining comparable computational overhead for 3D object detection. Inspired by them, we enhance the vanilla transformer to the mask-guided transformer, which exploits the foreground mask to guide the self-attention mechanism to explore semantic-level affinities. In this way, our transformer-based framework better matches the temporal action proposal task by learning semantic associations.

3 Method

Figure 2 illustrates the proposed framework, consisting of four stages: 1. Feature extractor (in Section 3.2) encodes the snippet-level features of the input video; 2. Foreground Mask Generate (FMG) module (in Section 3.3) exploits the global context to predict the foregrounds (action instances and boundaries) mask, representing the locations of the action-relevant frames throughout the video. 3.

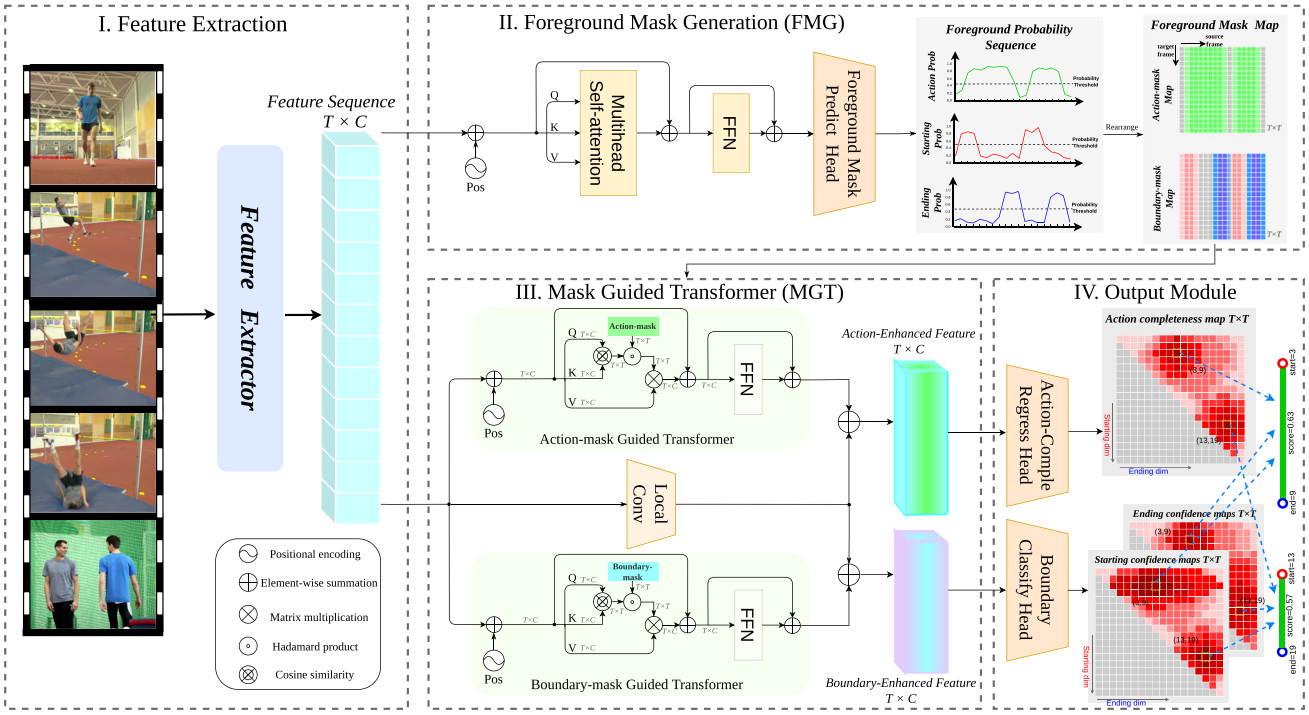

Fig. 2 Overview of our pipeline, which mainly includes four stages: 1. Feature extraction; 2. Foreground Mask Generation (FMG) module encodes the global context and predicts the foreground mask. 3. Mask-Guided Transformer (MGT) employs the foreground mask to learn semantic associations with foregrounds, cooperated with temporal convolution aggregates the local context to get the foregrounds-enhanced features. 4. Output modules predict action completeness and boundary confidence maps for proposal generation and evaluation

Mask-Guided Transformer (MGT) (in Section 3.4) exploits the foreground mask to calculate the semantic affinities with the foregrounds, extracting practical associations and suppressing background distractions. Meanwhile, temporal convolution assesses adjacent differences in fine-grain level and cooperates with MGT to get the foregrounds-enhanced features. 4. Output modules (in Section 3.5) predict the action completeness map and boundary confidence map for proposal generation and evaluation.

3.1 Notation and problem formulation

For an untrimmed raw video X, we can represent it as a frame sequence $X = \{x_n\}_{n=1}^{l_v}$ with l_v frames, where x_n is the n-th RGB frame of the video. Annotation of video X is composed by a set of action instances $\Psi_g = \{\varphi_n = (ts_n, te_n)\}_{n=1}^{N_g}$, where ts_n, te_n are the starting and ending time of the action instance φ_n, and N_g is the total number of ground-truth action instances. Temporal action proposal task is to generate a set of proposals $\Psi_p = \{\varphi_n = (ts_n, te_n)\}_{n=1}^{N_p}$ to cover ground-truth action instances in video X, where N_p is the number of proposals.

Since the TAP task requires the algorithms to locate the action instances' starting and ending times in a video on the temporal dimension, the main challenge is modeling temporal correlations between different timestamps. Specifically, action frames usually contain similar spatial appearances and motion patterns, while background frames always have semantic gaps with the action segments. Previous methods rarely capture these semantic-level associations, which is where we explore further towards this problem point. In this work, we mainly focus on learning semantic affinities between each frame x_n and action instances $\varphi_n = (ts_n, te_n)$ to extract high-quality temporal relations for the TAP task.

3.2 Feature encoding

Following previous TAP methods [22, 48], we adopt the two-stream network [33] to encode the video features to be fed into our model. Specifically, we first input the original untrimmed video into the feature extractor. Second, we split the untrimmed video frames $X = \{x_n\}_{n=1}^{l_v}$ into snippets sequence $S = \{s_n\}_{n=1}^{l_s}$ by a regular frame interval δ, where $l_s = l_v/\delta$. Then each snippet s_n is fed into the two-stream network to get the snippet feature $f_n \in \mathbb{R}^C$ of C-dimension. In this way, we can obtain the snippets feature sequence $F_s = \{f_n\}_{n=1}^{l_s}$. Finally, linear interpolation is adopted to maintain the same length of each video feature sequence fed into our model. So the feature extractor outputs the video features represented as $F \in \mathbb{R}^{T \times C}$ containing T snippets, which are shared by subsequent modules.

3.3 Foreground mask generation

The FMG module aims to generate foregrounds (action instances and action boundaries) masks, which are the locations of the action-relevant frames. Our FMG mainly consists of two submodules: a global-aware attention module (in Section 3.3.1) to build the long-range dependency along the video, and a foreground mask prediction head (in Section 3.3.2) for mask generation. We will introduce these two modules in the following sections.

3.3.1 Global-aware attention module

The global-aware attention module inputs the video features F and explores the interactions between snippets across the video. It has a standard one-layer transformer encoder architecture, which contains a positional-encoding, a multi-head self-attention, and an MLP with the residual connection. Given the video feature sequence F, we first add sinusoidal position encoding [37] to enable the transformer to obtain the frames' relative position in the video, forming the inputs F' of the transformer, formulated as:

$$F' = F + PE \tag{1}$$

$$\begin{aligned} PE(pos, 2i) &= sin(pos/10000^{2i/d}) \\ PE(pos, 2i+1) &= cos(pos/10000^{2i/d}) \end{aligned} \tag{2}$$

where pos is the temporal position, i is the dimension of snippet feature f_{pos}.

Then the multi-head self-attention module can learn different temporal interactions between snippets from different representation subspaces. Specifically, F' is first projected to three different subspaces by linear transformations, namely queries Q, keys K and values V, and then the multi-head self-attention feature $\hat{F}$ is calculated using the three projected representations:

$$Q = W_Q F', K = W_K F', V = W_V F' \tag{3}$$

$$\hat{F} = \mathrm{LN}(F' + softmax(\frac{QK^T}{\sqrt{d}})V) \tag{4}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{C' \times C}$ are learnable weights, $d = C/M$ indicates the dimension of each attention head, LN represents the layer normalization.

By adopting the self-attention mechanism, each snippet can learn its correlation with other snippets, so $\hat{F}$ contains the global context information. Next, to enhance the nonlinear characteristics of the attention feature, $\hat{F}$ is fed into an MLP with $ReLU$ activation function, residual connection, and layer normalization. Finally, the global-aware attention module outputs the feature $F^g \in \mathbb{R}^{T \times C}$

modeled by the global context for the attached mask prediction head.

3.3.2 Foreground mask prediction head

After modeling the global dependency, we employ the feature F^g to predict *action-instances foreground mask* and *action-boundaries foreground mask*. As shown in Fig. 3, we first evaluate each snippet's action, starting and ending probabilities to get the foreground probability sequences $\hat{S}^a, \hat{S}^s, \hat{S}^e \in \mathbb{R}^{T\times 1}$. We implement this for all snippets using stacked 1-D convolution layers H_m with $sigmoid$ activation on F^g:

$$\hat{S}^a, \hat{S}^s, \hat{S}^e = sigmoid(H_m(F^g)) \in \mathbb{R}^{T\times 1} \tag{5}$$

Second, we binarize the probabilities to get the foreground mask sequences $S^a, S^s, S^e \in \mathbb{R}^{T\times 1}$. The ith element $S_i \in \{0, 1\}$ in these sequences indicates the foregrounds binary probability of the ith snippet. We implement the binarization procedure with the $threshold$ function, formulated as:

$$S_i = \begin{cases} 1, & \hat{S}_i \geq \alpha_m \cdot max(\hat{S}) \\ 0, & \hat{S}_i < \alpha_m \cdot max(\hat{S}) \end{cases} \quad i = 1, 2, \cdots, T \tag{6}$$

where α_m is the foregrounds' binary probability threshold, we default the action mask threshold to 0.4 and the boundary mask threshold to 0.5. We ablate its value in Table 6.

Thirdly, since the attention map of $T \times T$ dimension in the MGT represents the correlations between any two snippets, we rearrange the 1D mask sequences $S^a, S^s, S^e \in \mathbb{R}^{T\times 1}$ into 2D mask maps $M^a, M^s, M^e \in \mathbb{R}^{T\times T}$ to guide the transformer to learn the semantic affinities. Specifically, we implement the rearrange procedure by repeating the mask sequences T times over the temporal dimension. In this way, the indexes with the mask value equal to 1 are the foreground's timestamps so that each frame could learn its association with action segments. We also ablate different formations of the rearranging process in the ablation study and Fig. 6.

Finally, we dilate these 2D mask maps to introduce the foregrounds' neighborhood information, which usually contains the changing trend of actions and is indispensable for action boundaries detection. We denote this process as:

$$M^a, M^s, M^e = \Phi(rearrange(S^a, S^s, S^e)) \in \mathbb{R}^{T\times T} \tag{7}$$

where M^a, M^s, M^e represent the predicted action, starting and ending foreground mask maps, respectively. The element $m_{i,j} \in \{0, 1\}$ in M represents the foreground binary probability of the jth snippet conditioned on the ith snippet. $\Phi(\cdot)$ denotes a kernel of size k used as a dilation operation. We experiment the ablation study of the kernel size in Table 7.

The mask prediction head outputs the foreground mask maps, which are employed later in the MGT to guide the transformer to build semantic-level affinities, enhancing the feature representation and further boost the action localization by learning the associations between frames and the actions.

3.4 Semantic affinities modeling

After getting the foreground masks, we employ them as prior knowledge to build the semantic associations with action segments. Specifically, we propose the mask-guided transformer (in Section 3.4.1) by exploiting the foreground mask maps to model semantic affinities with the action frames. Meanwhile, we employ the local convolution (in Section 3.4.2) to assess adjacent differences in the temporal dimension to achieve more fine-grained action boundaries detection. We will illustrate these two modules in the following sections.

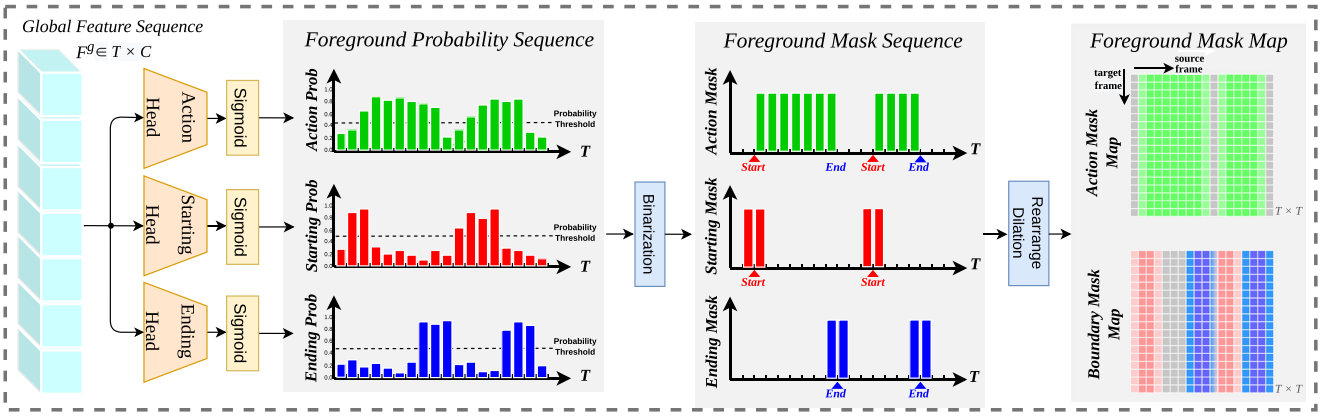

Fig. 3 The detailed process of the foreground mask generation. 1. First, evaluate each frame's action and boundary probabilities to get the foreground probability sequences; 2. Second, the probabilities are binarized to obtain the foreground mask sequences; 3. Finally, the rearrange and dilation operations are adopted to get the foreground mask maps

3.4.1 Mask-guided transformer

Lack of concentrated attention may lead to the failure to extract relevant information. Therefore, MGT exploits the foreground mask to guide the attention mechanism to build semantic associations, extracting action's features and suppressing the background distractions.

We designed two types of MGT to achieve the above goals: *Action Mask-Guided Transformer* and *Boundary Mask-Guided Transformer*. They take the video features and foreground masks as inputs, extracting semantic information under the guidance of the foreground masks. The Action-MGT employs the action mask M^a to learn the action persistence, while the Boundary-MGT captures the changing trend of actions with the guidance of the boundary masks M^s, M^e. Since their structures are in the same manner, we only illustrate the structure of the Boundary-MGT in Fig. 4.

Given the original video feature sequence F, we first generate the position-sensitive feature $\tilde{F}$ by adding the positional encoding to F. Then, we linear project $\tilde{F}$ to $\tilde{Q}, \tilde{V} \in \mathbb{R}^{T\times C'}$, where $\tilde{q}_i$ represents the query vector and $\tilde{v}_i$ represents the value vector for each snippet. To capture information from the starting and ending boundaries separately, we use two linear projections to generate $\tilde{K}^s, \tilde{K}^e \in \mathbb{R}^{T\times C'}$, representing the starting keys and the ending keys, respectively. Next, we calculate the cosine similarity of the queries and the keys to form the starting and

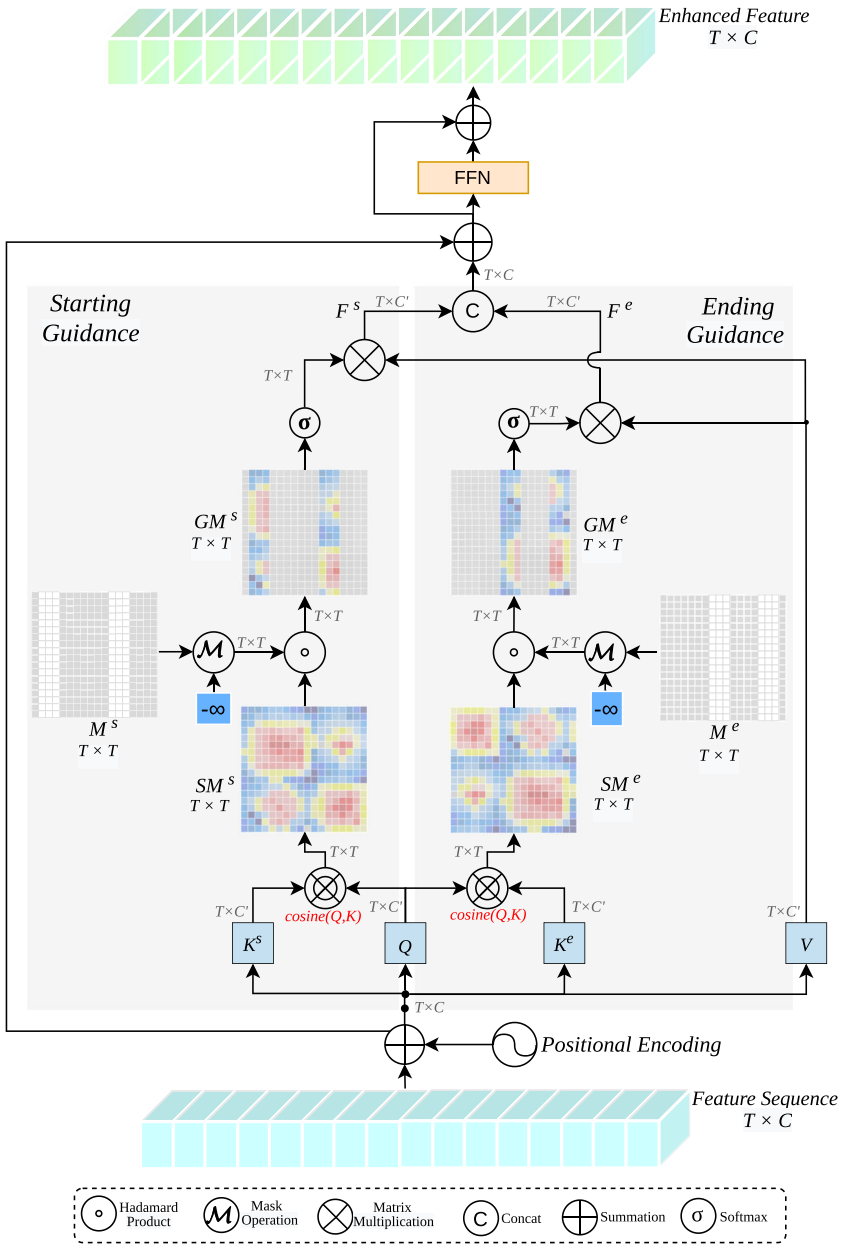

Fig. 4 The Mask-Guided Transformer (MGT) structure mainly consists of four steps: 1. Position encoding and linear projection to calculate cosine similarity map; 2. Performing Hadamard-product with foreground mask maps to obtain the mask-guided map; 3. Softmax normalization and multiplying with value matrix. 4. FFN and residual connection

ending similarity maps $SM^s, SM^e \in \mathbb{R}^{T\times T}$, formulated as:

$$SM^o = \frac{\tilde{Q}\cdot\tilde{K}^o}{|\tilde{Q}|\cdot|\tilde{K}^o|} \quad o=\{s,e\} \tag{8}$$

The above similarity maps establish the relationships between all snippets. However, this general global attention lacks the focalization of learning foreground action features. Meanwhile, the background nonaction snippets rarely contain valid information and even bring distractions. In order to concentrate attention on the foregrounds, we construct the masking operation $\mathcal{M}(\cdot)$ based on the foreground mask maps M^s, M^e, formulated as:

$$\mathcal{M}(M)_{ij} = \begin{cases} 1 & if \quad M_{ij}=1 \\ -\infty & if \quad M_{ij}=0 \end{cases} \tag{9}$$

We first perform this operation on the starting and ending mask maps, then hadamard-product the results with the similarity maps to get the mask-guided maps $GM^s, GM^e \in \mathbb{R}^{T\times T}$. By this means, the semantic affinities with the foreground segments are reserved in the mask-guided maps, and the other irrelevant elements are removed. This process is efficient in preserving contributive elements and suppressing noise, illustrated as follows:

$$GM^o = \mathcal{M}(M^o)\odot SM^o \quad o=\{s,e\} \tag{10}$$

where $\mathcal{M}(\cdot)$ represents the masking operation and $\odot$ denotes the hadamard-product.

Next, we apply the $softmax$ function to normalize the attention scores in the mask-guided maps, multiplying them with the value matrix $\tilde{V}$ to capture semantic associations and output the starting-enhanced and the ending-enhanced features $F^s, F^e \in \mathbb{R}^{T\times C'}$. Then, we concatenate the two types of features to aggregate information and use a convolution layer H_b to reduce the channel dimension from $2C'$ to C. Finally, residual connection and FFN are employed to enrich the nonlinearity, getting the boundary-enhanced feature $F^b \in \mathbb{R}^{T\times C}$, formulated as:

$$F^o = softmax(GM^o)\cdot\tilde{V} \quad o=\{s,e\} \tag{11}$$

$$\tilde{F^b} = H_b(\{F^s, F^e\}) \oplus \tilde{F} \tag{12}$$

$$F^b = FFN(\tilde{F^b}) \oplus \tilde{F^b} \tag{13}$$

where $\{\cdot,\cdot\}$ stands for the concatenation operation, $\oplus$ represents the element-summation, and FFN represents the feed forward network.

Throughout the above processes, the Action-MGT guides the action frames to extract more supportive cues, also instructing the nonaction frames to learn the semantic gap for better discrimination. The Boundary-MGT facilitates each frame to learn its similarity with action boundaries, helping boost the action boundary detection. As a result, the Action-MGT outputs the action-enhanced feature $F^a \in \mathbb{R}^{T\times C}$ and the Boundary-MGT outputs the boundary-enhanced feature $F^b \in \mathbb{R}^{T\times C}$ for following prediction heads.

3.4.2 Fine-grain context aggregation

Capturing the drastic change of actions in the short term is essential to detect the action starting and ending boundaries. Therefore, besides employing the MGT to model semantic associations, we also need to assess the differences of adjacent frames for more accurate boundary localization.

We employ the convolution on the temporal dimension to evaluate adjacent differences for fine-grained level boundary detection. Specifically, given the original video feature sequence $F \in \mathbb{R}^{T\times C}$, we exploit the 1D convolution layers with $kernel = 3$ to aggregate the local information and output the local-level features. Then, we fuse the local-level features with the foreground-enhanced features by summation to fuse the fine-level information and semantic-level associations. As a result, it outputs the action-enhanced fusion feature $F^{af} \in \mathbb{R}^{T\times C}$ for following action completeness regression, and the boundary-enhanced fusion feature $F^{bf} \in \mathbb{R}^{T\times C}$ for attached boundary classification.

3.5 Proposals generation and evaluation

After obtaining the foregrounds-enhanced features, following DBG [22], as shown in Fig. 5, we feed them to two output heads to predict the action completeness map and boundary confidence maps. In these score maps shown in Fig. 5, the row represents the starting dimension, and the column stands for the ending dimension. So each position $(i,j)_{i<j}$ represents the score of the proposal with $starting\ time = i$ and $ending\ time = j$. The product of the three scores of each proposal represents its confidence score for evaluation. We will illustrate the two output heads and the proposal generation and evaluation process in the following sections.

3.5.1 Action completeness regression head

The action completeness regression head receives the action-enhanced fusion feature F^{af} as input and outputs action completeness map P^c to estimate the IoU between candidate proposals and ground-truth action instances. First, it utilizes 1D convolution on F^{af} to generate the actionness score feature $P^a \in \mathbb{R}^{T\times 1}$, representing each snippet's action score. Second, following BSN [25] and DBG [22], by sampling features in each proposal's start, center, and end regions on P^a, it can construct the action completeness features FM^a for all candidate proposals. In this way, it transfers the P^a to three-dimensional proposal feature tensors $FM^a \in \mathbb{R}^{T\times T\times N}$, where $T\times T$ represents all the

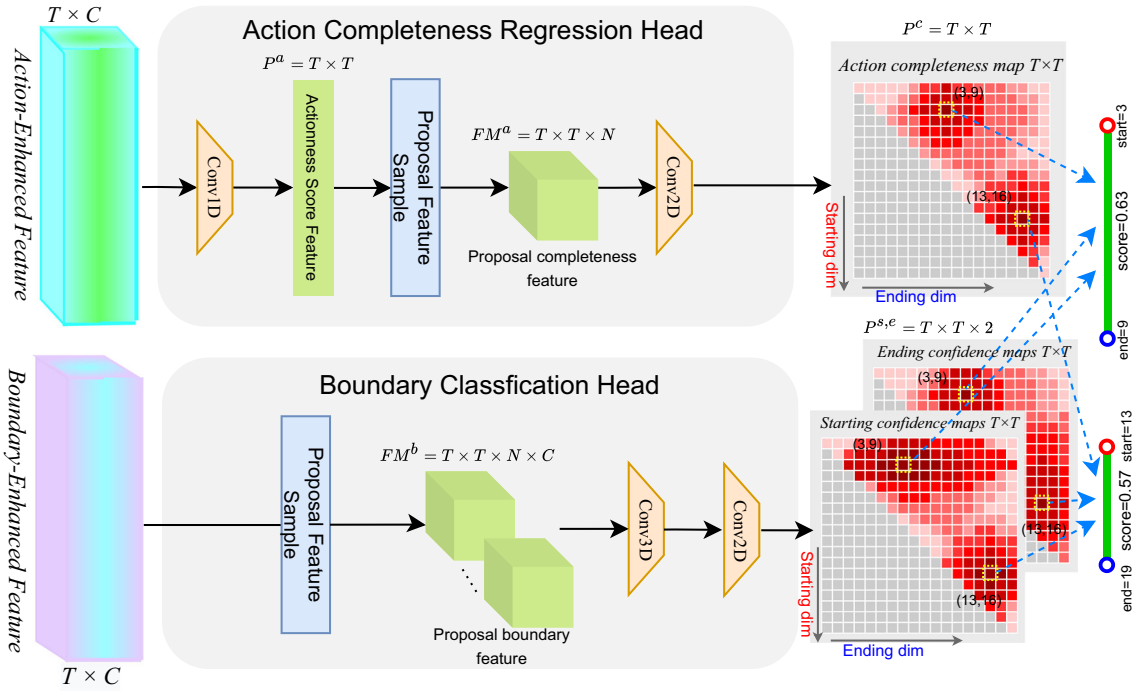

Fig. 5 Details of proposal generation and evaluation. It consists of an action completeness regression head and a boundary classification head

candidate proposals, N is the number of sampling points for each proposal and we set $N = 32$. Finally, it feds FM^a into a series of 2D convolution layers and the $sigmoid$ activation function to predict the action completeness map $P^c \in \mathbb{R}^{T\times T}$. These processes can be denoted as:

$$P^a = F_{(Conv1D)}(F^{af}) \tag{14}$$

$$FM^a = Sample(P^a) \tag{15}$$

$$P^c = Sigmoid(F_{(Conv2D)}(FM^a)) \tag{16}$$

In the action completeness map, each position $(i, j)_{i<j}$ represents the max IoU score between the proposal with $(t_s = i, t_e = j)$ and ground-truth action instances. We use the binary logistic regression loss to supervise P^a and the smooth L1 loss to supervise P^c during training.

3.5.2 Boundary classification head

The boundary classification head receives the boundary-enhanced fusion feature F^{bf} as input and outputs the boundary confidence maps $P^{s,e}$ to evaluate the action starting and ending probabilities for candidate proposals. First, it samples on the F^{bf} to construct four-dimensional proposal boundary features $FM^b \in \mathbb{R}^{T\times T\times N\times C}$, which contain $T \times T$ proposal features whose size is $N \times C$. We set $N = 32$ as the number of sample points and C as the channel numbers of the features. Then we utilize a 3D convolution layer to reduce the temporal dimension N and several 2D convolution layers to predict the boundary confidence maps $P^{s,e} \in \mathbb{R}^{T\times T\times 2}$. These steps can be written as follows:

$$FM^b = Sample(F^{bf}) \tag{17}$$

$$P^{s,e} = Sigmoid(F_{Conv2D}(F_{Conv3D}(FM^b))) \tag{18}$$

In the boundary confidence maps, rows represent the starting and columns represent the ending dimension. Each row in P^s represents the confidence score of the proposals with the same starting location, and each column in P^e represents the score of proposals with the same ending location. For the proposal (t_i, t_j) whose starting time is t_i and ending time is t_j, we multiply the scores of the corresponding position (i, j) in P^c, P^s, P^e as the confidence score for this proposal. Then we can select the top-K proposals with the highest confidence scores as our predictions.

3.6 Training

Given the annotation $\Psi_g = \{\varphi_n = (ts_n, te_n)\}_{n=1}^{N_g}$ of a video, for one of these ground-truth action instances $\varphi = (t_s, t_e)$, we define its action region as $r_g^a = [t_s, t_e]$, starting region as $r_g^s = [t_s - d_t, t_s + d_t]$ and ending region as $r_g^e =$

$[t_e - d_t, t_e + d_t]$, where d_t is the two temporal locations intervals.

Loss of the foregrounds mask generation FMG generates the action, starting and ending mask sequences $S^a, S^s, S^e \in \mathbb{R}^{T\times 1}$. For the i-th snippet s_i in the mask sequences, we denote its mask label as g_i. If foregrounds (action, starting or ending) exist in the snippet, we set $g_i = 1$, else we have $g_i = 0$. With three foregrounds-mask sequences, we can construct foreground mask generation loss using weighted binary logistic regression loss:

$$\mathcal{L}_{bl} = \sum_{i=1}^{T}(\alpha^+ \cdot g_i \cdot \log(s_i) + \alpha^- \cdot (1-g_i) \cdot \log(1-s_i)) \quad (19)$$

$$\mathcal{L}_{mask} = \mathcal{L}_{bl}(G^a, S^a) + \mathcal{L}_{bl}(G^s, S^s) + \mathcal{L}_{bl}(G^e, S^e) \quad (20)$$

where $\alpha^+ = T/\sum(g_i)$ and $\alpha^- = T/\sum(1-g_i)$ are balance factors.

Loss of the action completeness regression Following DBG [22], for the actionness score feature P^a, we calculate its binary logistic loss between G^a. And for action completeness map P^c, we denote the proposal corresponding to $p^c_{i,j}$ be $r_{i,j} = [i, j]$. We calculate the maximum Intersection-over-Union (IoU) between $r_{i,j}$ with all r^a_g to generate completeness label $g^c_{i,j}$, and adopt smooth L1 loss to construct the action completeness regression loss as:

$$\mathcal{L}_{comple} = \frac{1}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T} smooth_{L1}(p^c_{i,j} - g^c_{i,j}) + \mathcal{L}_{bl}(G^a, P^a) \quad (21)$$

Loss of the boundary classification For each location (i, j) within the boundary confidence maps $P^{s,e}$, we denote its starting region as $r^s_{i,j} = [i - d_t/2, i + d_t/2]$ and its ending region as $r^e_{i,j} = [j - d_t/2, j + d_t/2]$. Then we calculate the maximum overlap ratio IoR for $r^s_{i,j}$ with r^s_g, and $r^e_{i,j}$ with r^e_g to generate starting label $g^s_{i,j}$ and ending label $g^e_{i,j}$. We adopt the binary logistic regression to construct the starting classification loss and the ending classification loss:

$$\mathcal{L}_{start} = \frac{1}{T^2}\sum_{i=1}^{T}(g^s_{i,j} \cdot \log(p^s_{i,j}) + (1-g^s_{i,j}) \cdot \log(1-p^s_{i,j}))$$
$$\mathcal{L}_{end} = \frac{1}{T^2}\sum_{i=1}^{T}(g^e_{i,j} \cdot \log(p^e_{i,j}) + (1-g^e_{i,j}) \cdot \log(1-p^e_{i,j})) \quad (22)$$

The model is trained in the form of a multi-task loss function, with the overall loss function defined as:

$$\mathcal{L} = \mathcal{L}_{mask} + \lambda_1\mathcal{L}_{comple} + \lambda_2\mathcal{L}_{start} + \lambda_3\mathcal{L}_{end} \quad (23)$$

where λ_1, λ_2 and λ_3 are three scalars to balance the three terms, and defaulted as $\lambda_1, \lambda_2, \lambda_3 = 1$.

3.7 Inference

Score fusion With the boundary confidence maps $P^{s,e}$, each row in P^s represents the confidence score of the proposals with the same starting, and each column in P^e represents the score with the same ending. In order to obtain robust boundary confidence scores, we average the i-row in P^s and the i-column in P^e to represent the starting and the ending score of the temporal location t_i. In this way, we can get starting score sequence $p^s \in \mathbb{R}^{T\times 1}$ and ending score sequence $p^e \in \mathbb{R}^{T\times 1}$. Note that since the starting location is in front of the ending location, we only average the upper right part of the $P^{s,e}$.

Then, the element (i, j) in the action completeness map P^c represents the action completeness score of the proposal (t_i, t_j). So we fuse it with the boundary confidence scores to generate the final confidence score $P_{i,j}$:

$$P_{i,j} = P^c_{i,j} \times p^s_i \times p^e_j \quad (24)$$

Hence, we can get the dense candidate proposals set as $\Psi_p = \{\varphi_n = (t_i, t_j, P_{i,j})\}_{n=1}^{N}$

Post processing Because the above candidate proposals are generated by matching all starting and ending locations, resulting in redundant dense proposals. We adopt Soft-NMS to eliminate redundancy by a score decaying function. Then we can obtain the sparse candidate proposals set as $\Psi'_p = \{\varphi_n = (t_i, t_j, P'_{i,j})\}_{n=1}^{N'}$, where $P'_{i,j}$ is the final confidence score and N' is the final number of the candidate proposals.

4 Experiments and analysis

The purpose of our experiment is to evaluate the performance of our MGNet on the temporal action proposal task and the temporal action localization task, also to explore the principles behind the effects of semantic-level affinities modeling. Besides, we also perform ablation studies to validate the necessity and contribution of each component in our framework.

This section is organized as follows. The experimental targets of the TAP and TAL tasks and corresponding metrics are introduced in Section 4.1; Datasets and data preparation are described in Section 4.2; Implementation details are provided in Section 4.3; In Section 4.4 and Section 4.5, we compare our MGNet with the state-of-the-art methods on the TAP and TAL tasks, respectively. In Section 4.6, we conduct ablation studies to investigate the principles behind our framework; In Section 4.7, we visualize the qualitative results and analyze the performances. Finally, we

remark on the experimental contributions and limitations in Section 4.8.

4.1 Experimental tasks and metrics

Experimental tasks We experiment on two popular video understanding tasks to verify the effectiveness of our MGNet: Temporal Action Proposal (TAP) and Temporal Action Localization (TAL). The TAP task aims to detect the action instances' starting and ending times from untrimmed videos, while the TAL also requires determining the action categories.

Metrics of TAP TAP aims to generate proposals that overlap with ground-truth action instances at a high recall rate. According to the temporal Intersection over Union (tIoU) between proposals and ground-truth action instances, the predicted proposals can be divided into true positives, true negatives, false positives, and false negatives. Average Recall (AR) is defined as the ratio of true positives to the sum of true positives and false negatives, denoted as $AR = TP/(TP + FN)$. It represents the comprehensive capability of the model to detect all ground-truth instances. We calculate the AR@AN: Average Recall (AR) under different Average Number (AN) of proposals to evaluate proposals' quality. In addition, the Area Under the AR@AN Curve (AUC) is also applied for evaluation, which is more representative since it considers different AN simultaneously. Following the standard protocol, we use the tIoU thresholds set as [0.5:0.05:1.0].

Metrics of TAL We use the mean Average Precision (mAP) as the evaluation metric for the TAL task. A proposal is considered to be a true positive if its tIoU with the ground-truth instance is larger than a certain threshold and the predicted category is the same as this ground-truth instance. Average Precision (AP) is defined as the ratio of true positives to the sum of true positives and false positives, denoted as $AP = TP/(TP + FP)$. It represents the model's detection accuracy. We calculate the mAP at the tIoU threshold set as {0.3,0.4,0.5,0.6,0.7}, and we also report the average mAP of all the tIoU thresholds.

4.2 Datasets and preparation

THUMOS14 [21] is a standard benchmark for action localization. Its training, validation, and testing sets contain 13320, 1010, and 1574 untrimmed videos, respectively. The temporal action localization task of THUMOS14, which contains videos over 20 hours from 20 sports classes, is very challenging since the duration of action instances varies a lot throughout the videos. Following common settings, we train and test our model with temporal annotated videos in this dataset, that is, 200 untrimmed videos from the validation set and 213 from the testing set.

ActivityNet-1.3 [6] is a large-scale action understanding dataset that contains 19994 temporal annotated untrimmed videos with 200 action categories. The training, validation, and testing sets are divided into 2:1:1. Each video has an average of 1.65 action instances. Following the standard practice, we train our model with the training set and evaluate its performance on the validation set.

Data preparation Since the datasets are long and untrimmed videos, which usually contain thousands of frames, extracting the video features online is very costly and time-consuming. Following the usual practice [22], we employ the pre-trained action recognition network to extract the video feature offline and take them as the input of our model.

For THUMOS14, the video features are extracted using the TSN model [39] pre-trained on Kinetics [8] with the snippet interval $\delta = 5$. And we crop each video feature sequence by sliding window with $Length = 128$ and $stride = 64$.

For ActivityNet-1.3, we adopt the two-stream network pre-trained on ActivityNet-1.3 by Xiong et al. [46] with $\delta = 16$ to extract the feature representation. Furthermore, we rescale the feature sequences to $T = 100$ by linear interpolation to form the feature sequences fed into our model.

4.3 Implementation details

Parameters and steps The hyperparameters are empirically defined as follows: In the global-aware attention module, the transformer encoder's layer is set to 1 to avoid overfitting, and the numbers of multi-head self-attention heads are set to 8. To exploit the semantic-level affinities, the numbers of the mask-guided transformer layer is also set to 1. For the foregrounds' binary probability threshold, we set $\alpha_m = 0.4$ for the action mask and $\alpha_m = 0.5$ for the boundary mask. The dilation kernel size of the foreground mask map is set to $k = (3, 3)$. The soft-NMS threshold is set to 0.8 and 0.65 on ActivityNet-1.3 and THUMOS14, respectively.

The steps of implementing the TAP and TAL tasks using our model are as follows: First, we employ the pre-trained TSN model [39] to extract the videos' RGB features and optical flow features. Second, following DBG [22], we feed the two types of features into parallel mask-guided transformers to mine the semantic associations simultaneously. Then we fuse the two features by summation and input them into the prediction heads to generate action proposals with corresponding confidence

scores. Next, soft non-maximum suppression (Soft-NMS) is adopted to suppress the redundant proposals and implement the TAP task. To achieve the TAL task, we finally keep the top-200 proposals, and take the top-2 video-level classification labels and corresponding scores from UntrimmedNet [38], multiplying them by the confidence scores to generate action localization results.

Training We train our model using the Adam optimizer, the learning rate is set to 10^{-3} for the first 8 epochs, and decayed by a factor of 0.1 for another 2 epochs. The weight decay is set to 10^{-4}, the dropout ratio in the mask-guided transformer is 0.1. We train our model on a single 2080Ti GPU with batch size 16.

Testing We first generate action proposals with confidence scores using our MGNet. Then Soft-NMS is employed to suppress redundant proposals, we set the threshold to 0.8 and 0.65 on ActivityNet-1.3 and THUMOS14, respectively. Finally, we multiply the classification scores from UntrimmedNet with the confidence scores as the final scores for calculating mAP.

4.4 Evaluation of the TAP task

We compare our MGNet with recent state-of-the-art methods for the temporal action proposal task on the THUMOS14 dataset and ActivityNet-1.3 dataset to verify the superior performance of our model.

4.4.1 Performance on THUMOS14 dataset

Table 1 demonstrates the temporal action proposal performance comparison on the testing set of THUMOS14. To ensure a fair comparison, we adopt C3D and two-stream features to conduct the comparison experiment like the previous methods [2, 22]. Experiment results suggest that our MGNet outperforms other local-level, proposal-level, and global-level methods with both C3D and two-stream features. In addition, NMS improves the average recall rate under small proposal numbers, while soft-NMS performs better when AN is higher. It proves that our method could learn more effective information by mining semantic-level affinities, achieving more accurate action localization by extracting efficient associations.

4.4.2 Performance on ActivityNet-1.3 dataset

We compare our MGNet for proposal generation performance on the ActivityNet-1.3 validation set. Table 2 lists a set of state-of-the-art methods, including BSN [25], BMN [23], BC-GNN [2], DBG [22], RapNet [16], TCA-Net [31] and RTD-Net [35]. The result shows that our MGNet outperforms other methods, especially improving AUC from 68.23% to 68.85%, demonstrating that our model has good results under different AN simultaneously and achieves an overall performance promotion of action proposal generation.

Furthermore, on both ActivityNet-1.3 and THUMOS14 datasets, our model outperforms the concurrent TCA-Net [31] based on the attention mechanism and the RTD-Net [35] with transformer-alike architecture. TCA-Net employs attention to aggregate global interactions, but instinctively employing the attention on the global perspective lacks concentration on foreground frames. RTD-Net uses the transformer decoder to implement the set prediction of proposals. However, it does not fully utilize the transformer encoder's global dependency modeling ability. Our MGNet employs the transformer to model semantic relations with action segments. It has a more robust, high-quality feature mining ability and performs better.

4.4.3 Comparison and analysis

Compared with the *local-level* methods such as TURN [17] and BSN [25], they only mine the local context, which is usually insufficient. Our model obtains more temporal information by establishing the foreground interactions. Compared with BMN [23], BC-GNN [2], and DBG [22], which adopt GCNs or the sampling strategy to construct *proposal-level* feature correlations, our MGNet adopts the mask-guided transformer to build semantic-level relations, which are more representative and efficient. The *global-level* methods RapNet [16], TCA-Net [31], and RTD-Net [35], build long-range dependencies using the vanilla vision transformer [1] or the query-and-retrieval procedure. These methods lack the focalization to learn the action's valid information from the video and introduce background distractions. However, our MGNet exploits the foreground mask as prior knowledge to model semantic associations with action frames, efficiently extract high-quality information and suppress background distractions.

As a result, our MGNet significantly outperforms other methods on both THUMOS14 and ActivityNet-1.3 datasets for the temporal action proposal task. Thus, it proves that exploiting semantic-level affinities with our MGNet is effective and efficient.

4.5 Evaluation of the TAL task

To evaluate the quality of proposals, we test the proposals generated by our MGNet on the temporal action localization task. We calculate the mean Average Precision (mAP) under different IoU thresholds to evaluate action localization performance. We set the IoU thresholds as {0.3,0.4,0.5,0.6,0.7} for THUMOS14.

Table 1 Compare our model with other state-of-the-art methods for the TAP task on the THUMOS14 dataset

Feature	Method Types	Models	AR@AN				
			@50	@100	@200	@500	@1000
C3D	Local-level	SCNN-prop [32]	17.22	26.17	37.01	51.57	58.20
		SST [5]	19.90	28.36	37.90	51.58	60.27
		TURN [17]	19.63	27.96	38.34	53.52	60.75
		BSN [25]+NMS	27.19	35.38	43.61	53.77	59.50
		BSN [25]+SNMS	29.58	37.38	45.55	54.67	59.48
		MGG [27]	29.11	36.31	44.32	54.95	60.98
	Proposal-level	BMN [23]+NMS	29.04	37.72	46.79	56.07	60.96
		BMN [23]+SNMS	32.73	40.68	47.86	56.42	60.44
		DBG [22]+NMS	32.55	41.07	48.83	57.58	59.55
		DBG [22]+SNMS	30.55	38.82	46.56	56.42	62.17
		BC-GNN [2]+NMS	33.56	41.20	48.23	56.54	59.76
		BC-GNN [2]+SNMS	33.31	40.93	48.15	56.62	60.41
	Gloabl-level	RapNet [16]	29.72	37.53	45.61	55.26	61.32
		CAN [26]	30.79	38.39	47.59	56.02	61.44
	Semantic-level	MGNet(Ours)+NMS	**34.29**	**42.75**	**50.53**	**59.28**	62.02
		MGNet(Ours)+SNMS	31.50	40.10	48.24	57.69	**62.88**
2-stream	Local-level	TAG [58]	18.55	29.00	39.61	-	-
		TURN [17]	21.86	31.89	43.02	57.63	64.17
		CTAP [15]	32.49	42.61	51.97	-	-
		BSN [25]+NMS	35.41	43.55	52.23	61.35	65.10
		BSN [25]+SNMS	37.46	46.06	53.21	60.64	64.52
		MGG [27]	39.93	47.75	64.65	61.36	64.06
	Proposal-level	BMN [23]+NMS	37.15	46.75	54.84	62.19	65.22
		BMN [23]+SNMS	39.36	47.72	54.70	62.09	65.49
		DBG [22]+NMS	40.89	49.24	55.76	61.43	61.95
		DBG [22]+SNMS	37.32	46.67	64.50	62.21	66.40
		BC-GNN [2]+NMS	41.15	50.53	56.23	61.45	66.00
		BC-GNN [2]+SNMS	40.50	49.60	56.33	62.80	66.57
	Gloabl-level	RapNet [16]	40.35	48.23	54.92	61.41	64.47
		CAN [26]	41.33	47.99	55.42	62.43	64.91
		RTD-Net [35]+I3D features	41.52	49.32	56.41	62.91	-
		TCA-Net [31]	42.05	50.48	57.13	63.61	66.88
	Semantic-level	MGNet(Ours)+NMS	**45.40**	**53.05**	**59.05**	63.81	64.19
		MGNet(Ours)+SNMS	41.36	51.04	58.36	**65.04**	**68.68**

The metrics are AR@AN. SNMS stands for Soft-NMS

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

Table 2 Performance comparison with state-of-the-art proposal generation methods on the validation set of ActivityNet-1.3 in terms of AR@AN and AUC

Method	BSN [25]	BMN [23]	BC-GNN [2]	DBG [22]	RapNet [16]	TCA-Net [31]	RTD-Net [35]	MGNet(Ours)
AR@100(val)	74.16	75.01	76.73	76.65	76.71	76.08	73.21	**77.12**
AUC(val)	66.17	67.10	68.05	68.23	67.63	68.08	65.78	**68.85**

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

4.5.1 Detection strategy

We adopt the two-stage "detection by classifying proposals" TAL framework, which feeds the predicted proposals to the action classifier for action recognition, thus implementing temporal action localization. Specifically, we first generate a set of action proposals with confidence scores with our model, and utilize the Soft-NMS to keep top-200 proposals. Then we take the top-2 video-level classification labels and corresponding scores from UntrimmedNet [38], multiplying them by the confidence scores for detection evaluation.

4.5.2 Comparison and analysis

Table 3 lists some state-of-the-art temporal action localization methods. We use the same TSN features [39] for a fair comparison, except RTD-Net employs the I3D features [8]. In addition, we also use the same action classifier UntrimmedNet [38].

Our MGNet achieves an average mAP of 48.4% ([0.3:0.1:0.7]), with an mAP of 50.1% at tIoU=0.5 and an mAP of 28.3% at tIoU=0.7, implying that our method can recognize and localize actions much more accurate than any other method. Note that our method also outperforms the concurrent works of RTD-Net [35] and TCA-Net [31], which also employ the transformer architecture and attention mechanism for proposal detection. Different from them, our model employs the foreground mask as prior knowledge to guide the transformer to learn the semantic-level affinities with the action frames. By this means, our transformer can focus on and extract more critical information from contributive frames, modeling robust feature representations and achieving better performances.

4.6 Ablation study

To verify the effectiveness of our method, we conduct the following ablation studies: we ablate different forms the network architectures, such as the formations of the foreground mask (Section 4.6.1) and the fusion strategy of the MGT and convolution (Section 4.6.2), so as to investigate the contribution of each component. We also ablate some hyperparameters, such as the threshold of binarization probability (Section 4.6.3), the size of the foreground mask dilation kernel (Section 4.6.4), and the number of MGT's layers (Section 4.6.5), to find the best parameter settings.

4.6.1 The foreground mask formations

Our MGT learns the semantic-level affinities and extracts informative features from action segments by utilizing the foreground mask map, which is the core component of our model. So we set up various formations of the action-instance mask maps to compare and analyze the effects of different formations of masks.

As shown in Fig. 6, we compare five different formations of the foreground mask maps. The first *w/o mask* means

Table 3 Comparison between our model with other temporal action localization state-of-the-art methods on THUMOS14 testing set in terms of mAP(%)

Method	0.7	0.6	0.5	0.4	0.3	Avg.
SST [4]	4.7	10.9	20.0	31.5	41.2	21.7
TURN [17]	6.3	14.1	24.5	35.3	46.3	25.3
BSN [25]	20.0	29.4	36.9	45.0	53.5	36.8
MGG [27]	21.3	29.5	37.4	46.8	53.9	37.8
BMN [23]	20.5	29.7	38.8	47.4	56.0	38.5
DBG [22]	21.7	30.2	39.8	49.4	57.8	39.8
G-TAD [48]	23.4	30.8	40.2	47.6	54.5	39.3
BC-GNN [2]	23.1	31.2	40.4	49.1	57.1	40.2
RTD-Net [35]	25.0	36.4	45.1	53.1	58.5	43.6
TCA-Net [31]	26.7	36.8	44.6	53.2	60.6	44.4
CAN [26]	22.4	30.3	39.7	48.7	57.9	39.8
Gemini [61]	21.4	32.5	42.6	50.6	56.7	40.8
A2Net [50]	17.2	32.5	45.5	54.1	58.6	41.6
KFC [10]	23.8	33.6	44.9	52.7	59.3	42.9
Xia et al. [45]	24.0	33.5	44.2	52.2	61.9	43.2
MGNet(Ours)	**28.3**	**38.7**	**50.1**	**59.3**	**65.8**	**48.4**

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

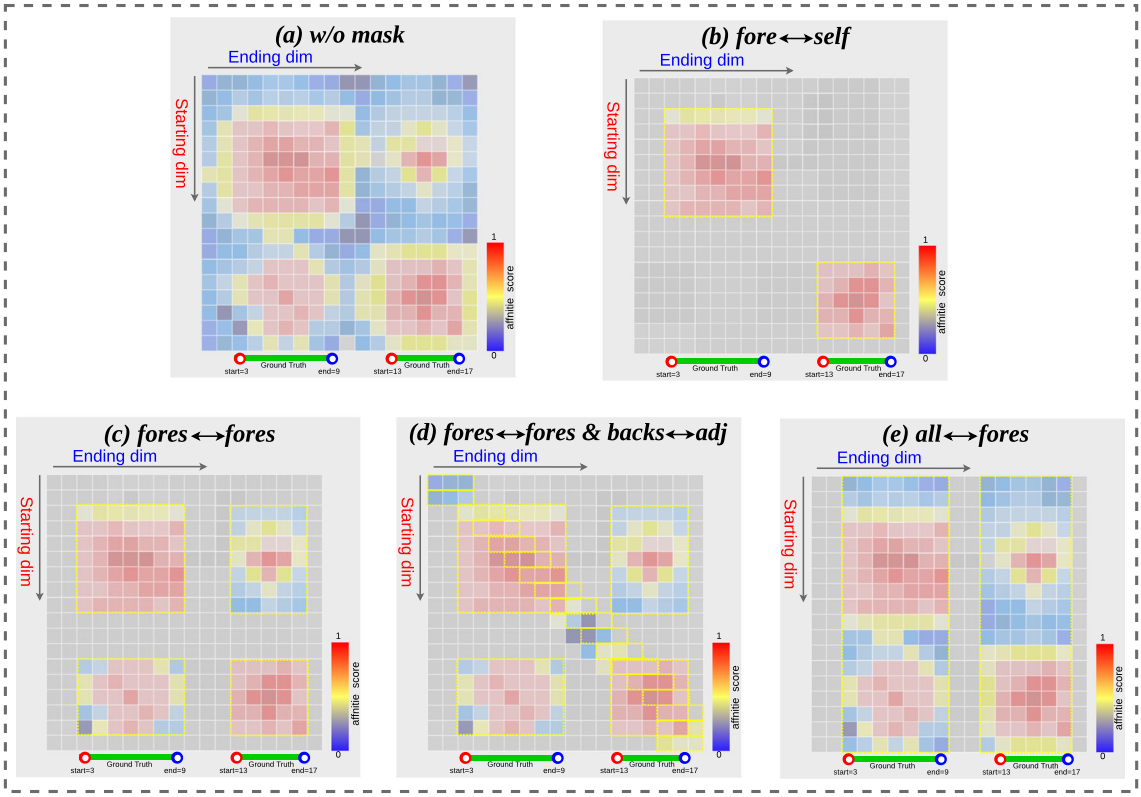

Fig. 6 Ablation in the different formations of the action-instance foreground mask maps

it does not adopt the foreground mask to guide the MGT to learn semantic associations. The second *fore↔self* mask formation means each action segment extract features from itself, so it lacks learning supportive cues from other co-occurring action frames. The third *fores↔fores* formation represents that the action segments build semantic relations with each other. In this way, the connection among different action instances in the same video can be established. The fourth *fores↔fores & backs↔adj* means that the action segments capture the correlations with each other, and the nonaction frames aggregate features from adjacent frames. The last *all↔fores* mask formation represents that all frames exploit the semantic-level affinities with foreground action segments.

Table 4 shows the experimental results of different formations of the foreground mask maps. Comparing the *w/o mask* of the first row with the others, we can see that the transformer is less effective without the guidance of the foreground mask since it is not guided to focus on the essential foregrounds, lacking focalization to capture semantic information from the action frames. In addition, the transformers' intensity of extracting representations from valid video scenarios is weakened due to the introduction of background distractions. By comparing the second row with the third row, we can conclude that the supportive cues between action segments can help extract more action clues. It means that the correlations between different action instances can be

Table 4 Performance comparison of the different formations of action-instance foreground mask maps on ActivityNet-1.3

Formations of foregrounds mask map	AR@10	AR@100	AUC
w/o mask	57.56	76.64	68.41
fore↔self	57.29	76.70	68.40
fores↔fores	57.43	76.71	68.52
fores↔fores & backs↔adj	57.40	77.02	68.64
all↔fores	**57.93**	**77.12**	**68.85**

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

The italic entries represent the abbreviated names of different methods in our ablation experiments

Table 5 Performance analysis of different fusion strategies of the MGT and the convolution on ActivityNet-1.3

	MGT	Conv	Fusion Strategy	AR@10	AR@100	AUC
1	✓		-	57.09	76.56	68.35
2		✓	-	57.50	76.36	68.15
3	✓	✓	MGT in series with Conv	56.35	75.53	67.30
4	✓	✓	Conv in series with MGT	56.69	76.05	67.70
5	✓	✓	MGT parallels with Conv by concatenation	57.81	76.83	68.61
6	✓	✓	MGT parallels with Conv by summation	**57.93**	**77.12**	**68.85**

Conv strands for convolution

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

established since the actions are mostly related in a specific video. The last row *all*↔*fores* performs best since it allows all source frames to mine the semantic-level affinities with the foreground action clips. In this way, foreground frames can build correlations and connections between different action instances. Meanwhile, background frames can discover the semantic gap between themselves and action instances, helping better distinguish background distractions. Therefore, we adopt this formation of rearranging to construct our foreground mask map.

4.6.2 Fusion strategies of the MGT and convolution

Our MGT exploits the foreground mask to guide the transformer to capture semantic associations, while fine-grained convolution assesses adjacent differences for precise boundary detection. We explore the different fusion strategies of the MGT and convolution in Table 5. It can be seen that without MGT or convolution, both lead to performance degradation. In particular, the results are lower without the MGT, proving that our mask-guided transformer extract more informative representations by modeling semantic affinities. Moreover, we explore the different fusion strategies of the MGT and convolution, including series and parallel connection. The *MGT in parallel with Conv using summation* outperforms other fusion strategies.

4.6.3 Binary probability threshold

During predicting the foreground mask maps, as shown in Fig. 3, we transform the foreground probability sequences into the mask sequences by employing the $threshold$ binarization function. Different thresholds α_m will generate different foreground mask maps. Table 6 ablates the action-foreground and boundary-foreground threshold on the THUMOS14 testing set with the C3D feature.

Comparing the results under different α_m settings, we observe the best performance improvement when the action's probability threshold $\alpha_m = 0.4$ and the boundary's threshold $\alpha_m = 0.5$. We believe that when the threshold is set too low, false positives and background interferences will be introduced, distracting the MGT from learning semantic associations from foreground segments. However, when the threshold is set too high, MGT can only extract less action information, which is insufficient to support temporal modeling. Therefore, we conclude that a moderate threshold setting can boost our model more accurate and efficient.

4.6.4 Dilation kernel size

We dilate the foreground mask map to introduce the foregrounds' surrounding information, which usually contains the changing trends of actions in the video. Furthermore,

Table 6 Ablation study on the binary probability threshold of the action foregrounds and boundary foregrounds on THUMOS14

Probability threshold		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Action	AR@50	42.40	42.92	43.96	**45.40**	44.40	44.92	44.82	42.37	43.45
	AR@100	50.80	51.40	51.90	**53.05**	52.15	52.54	52.30	51.03	51.64
Boundary	AR@50	43.06	43.16	44.51	44.31	**45.40**	43.45	42.68	42.68	42.56
	AR@100	51.03	51.42	52.12	52.35	**53.05**	51.14	50.80	51.37	50.93

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

Table 7 Ablation study on the dilation kernel size of the foreground mask map on THUMOS14

Kenel size k	AR@50	AR@100	AR@200
1	44.21	52.50	58.67
3	**45.40**	**53.05**	**59.05**
5	44.26	52.36	58.10
7	42.85	51.55	52.97
9	42.66	51.43	57.95

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

since the predicted foreground mask may be biased in the temporal dimension, the dilation process can reintroduce the false-negative snippets for the MGT to extract their valid representations. Table 7 shows the results under different dilation kernel size k on the THUMOS14 dataset. From the table, we can see that $k = 3$ performs best. The results are not good when $k = 1$ since it lacks capturing the moving trends from around the action boundaries. It also does not reintroduce the false-negative frames to extract their effective features. On the contrary, excessive negative nonaction frames will be introduced when the dilation kernel size is larger, weakening the foreground mask's guidance effect.

4.6.5 Number of transformer layers

The previous transformer-based approaches used for NLP [37] or Vision [7] demonstrate that stacking more layers often brings better performance. We stack different numbers of transformer layers in our MGT for comparison. Table 8 shows that our MGT performs best in shallow layers. We believe this is mainly caused by the distribution of the dataset and the degree of the model's overfitting. Specifically, the ActivityNet-1.3 dataset contains about 20K videos, but its training set videos mainly include one only action instance or several action segments of the same class. With a small number of samples of different categories, our mask-guided transformer with deeper layers easily converges to a local optimum. As for the THUMOS14 dataset, each video has various action instances of different classes. However, with only 200 annotated videos available to train our network in this dataset, our deeper MGT is prone to over-fitting. Therefore, the number of our MGT layers set to 1 is the most appropriate. We strive to achieve better performance in the future by acquiring more datasets or using some data augmentation methods to train a deeper network.

Table 8 Ablation study on the different number of transformer layers in MGT on ActivityNet-1.3

No. of transformer layers	AR@10	AR@100	AUC
1	**57.93**	**77.12**	**68.85**
2	57.46	76.61	68.39
3	57.57	76.48	68.33
6	57.25	76.44	68.21

The bold entries represent the better results in our comparison experiments for readers convenient to read and compare

4.7 Visualization and analysis

Figure 7 visualizes some representative action localization results for the videos with different challenge cases. For each video with n ground-truth, we choose the proposals with confidence score top-n for visualization.

In the first video, the person doing the action and the action object are present in both background and foreground segments. Specifically, they both contain a man holding a saxophone. However, he assembles the saxophone in the background clips while playing the saxophone only in the foreground. Our model accurately learned the interaction between the actor and the action object, helping detect the interval of playing saxophone.

The challenge of the second video is that an indistinct background clip exists between two action segments. Specifically, the athlete first exercises on the parallel bars. Then, he jumps off for a short rest next to the parallel bars and later goes back for exercise again. This short action pause is hard to distinguish and may easily be misclassified as a false positive. Our MGNet learns the semantic gap between the short rest clip and the action frames to suppress the false positive detections. As a result, our top-2 proposals accurately detect these two action segments and identify the background clip between the two foregrounds, proving that modeling the semantic-level affinities is effective and efficient.

The difficulty of the third video is that multiple action clips exist in the video, so our top-n proposals need to detect all ground-truth action instances accurately at the same time. We visualize the results of our top-4 proposals, which all precisely localize the action instances and distinguish the backgrounds. In summary, our MGNet can handle videos in different complex scenes with strong robustness and high accuracy.

4.8 Contribution and limitation

Contribution Based on the experiments and analysis above, we remark the contribution of this paper as follows. We propose the MGNet for temporal action proposal in videos. It learns the semantic-level affinities between action frames to enhance the feature representations and further boost the action localization. In the experiment, we first compare our MGNet with other advanced methods on the TAP and TAL

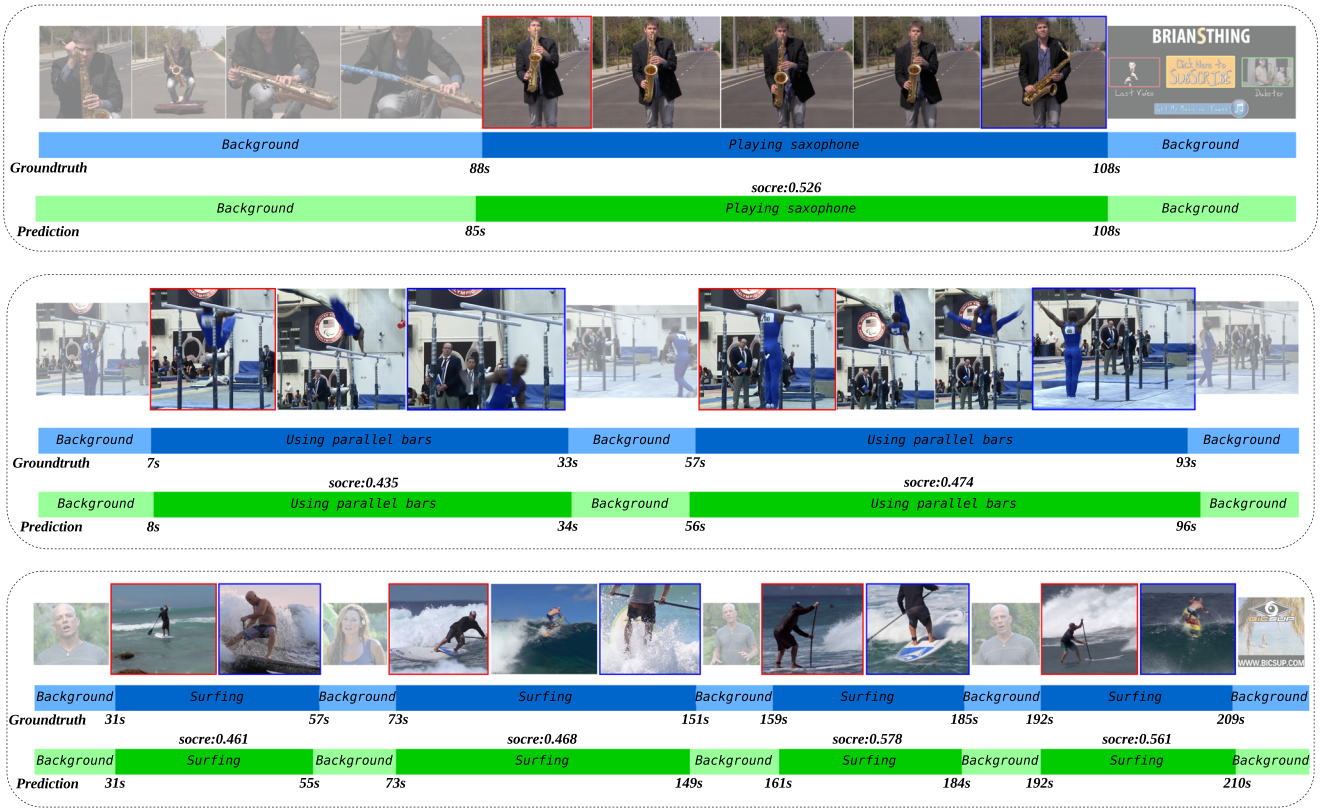

Fig. 7 Visualization examples of temporal action localization with the proposals generated by our model

tasks, verifying the effectiveness of our model. Second, we conduct the ablation studies on our model. We experiment with different formations of the foreground mask map to analyze its principles, and remove the MGT to prove its necessity. Results show that each component contributes to accurate action localization.

Limitation The experiment limitation is that utilizing the video features pre-extracted offline decreases the model's efficiency, and the feature representations also influence the detection results. Specifically, since the datasets in the TAP task are long untrimmed videos that usually contain thousands of frames, extracting video features online is costly and time-consuming. Therefore, the usual practice is to employ the pre-trained action recognition network to extract the video features offline, and then take them as the input of the model. In this case, the model's performance is affected by the input features. As shown in Table 1, both previous methods and our model perform better using the two-stream features than the C3D features. Therefore, in future work, we will focus on designing an end-to-end framework to efficiently extract the video features in real-time with lower-cost resources for the temporal action localization task.

5 Conclusion

This paper proposes a novel framework called MGNet for temporal action proposal in videos. MGNet exploits the foreground mask as prior knowledge to model semantic-level associations with action segments, enhancing feature representations and further boosting action localization. First, we design the Foreground Mask Generation (FMG) module to generate the foreground mask, representing the locations of the action-related frames across the video. Then we propose a Mask-Guided Transformer (MGT) by exploiting the foreground mask to guide the transformer to learn the semantic-level affinities, building intra-semantic similarities for foregrounds to extract supportive cues from co-occurring actions, and modeling the inter-semantic gaps between backgrounds and action frames for better distinction. Extensive experiments conducted on ActivityNet-1.3 and THUMOS14 demonstrate that our model can achieve superior performance on both TAP and TAL tasks. In the future, to improve the model's learning efficiency, we will further explore an end-to-end network to extract the video features and model the temporal relations simultaneously, implementing an online real-time action localization method.

References

1. Arnab A, Dehghani M, Heigold G, Sun C, Lučić M, Schmid C (2021) Vivit: a video vision transformer. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6836–6846
2. Bai Y, Wang Y, Tong Y, Yang Y, Liu Q, Liu J (2020) Boundary content graph neural network for temporal action proposal generation. In: European conference on computer vision. Springer, pp 121–137
3. Bertasius G, Wang H, Torresani L (2021) Is space-time attention all you need for video understanding? In: ICML, vol 2, p 4
4. Buch S, Escorcia V, Ghanem B, Fei-Fei L, Niebles JC (2017) End-to-end, single-stream temporal action detection in untrimmed videos. In: Procedings of the British machine vision conference 2017. British machine vision association, pp 93–93
5. Buch S, Escorcia V, Shen C, Ghanem B, Carlos Niebles J (2017) Sst: single-stream temporal action proposals. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 2911–2920
6. 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
7. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European conference on computer vision. Springer, pp 213–229
8. 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
9. Chen W, Chai Y, Qi M, Sun H, Pu Q, Kong J, Zheng C (2022) Bottom-up improved multistage temporal convolutional network for action segmentation. Appl Intell, pp 1–17
10. Ding X, Wang N, Gao X, Li J, Wang X, Liu T (2021) Kfc: an efficient framework for semi-supervised temporal action localization. IEEE Trans Image Process 30:6869–6878
11. Du Z, Mukaidani H (2022) Linear dynamical systems approach for human action recognition with dual-stream deep features. Appl Intell 52(1):452–470
12. Duke B, Ahmed A, Wolf C, Aarabi P, Taylor GW (2021) Sstvos: sparse spatiotemporal transformers for video object segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5912–5921
13. Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6202–6211
14. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1933–1941
15. Gao J, Chen K, Nevatia R (2018) Ctap: complementary temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV), pp 68–83
16. Gao J, Shi Z, Wang G, Li J, Yuan Y, Ge S, Zhou X (2020) Accurate temporal action proposal generation with relation-aware pyramid network. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp 10810–10817
17. Gao J, Yang Z, Chen K, Sun C, Nevatia R (2017) Turn tap: temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE international conference on computer vision, pp 3628–3636
18. Gao L, Li T, Song J, Zhao Z, Shen HT (2020) Play and rewind: context-aware video temporal action proposals. Pattern Recogn 107477:107
19. Gao Y, Liu X, Li J, Fang Z, Jiang X, Huq KMS (2022) Lft-net: local feature transformer network for point clouds analysis. IEEE transactions on intelligent transportation systems
20. Jiang G, Jiang X, Fang Z, Chen S (2021) An efficient attention module for 3d convolutional neural networks in action recognition. Appl Intell 51(10):7043–7057
21. Jiang YG, Liu J, Zamir AR, Toderici G, Laptev I, Shah M, Sukthankar R (2014) Thumos challenge: action recognition with a large number of classes
22. Lin C, Li J, Wang Y, Tai Y, Luo D, Cui Z, Wang C, Li J, Huang F, Ji R (2020) Fast learning of temporal action proposal via dense boundary generator. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 11499–11506
23. 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/CVF international conference on computer vision, pp 3889–3898
24. Lin T, Zhao X, Shou Z (2017) Single shot temporal action detection. In: Proceedings of the 25th ACM international conference on Multimedia, pp 988–996
25. 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 (ECCV), pp 3–19
26. Liu Y, Chen J, Chen X, Deng B, Huang J, Hua XS (2021) Centerness-aware network for temporal action proposal. IEEE Trans Circuits Syst Video Technol 32(1):5–16
27. Liu Y, Ma L, Zhang Y, Liu W, Chang SF (2019) Multi-granularity generator for temporal action proposal. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3604–3613
28. Mao J, Xue Y, Niu M, Bai H, Feng J, Liang X, Xu H, Xu C (2021) Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3164–3173
29. Neimark D, Bar O, Zohar M, Asselmann D (2021) Video transformer network. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3163–3172
30. Pérez-Hernández F., Tabik S, Lamas A, Olmos R, Fujita H, Herrera F (2020) Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: application in video surveillance. Knowl-Based Syst 105590:194
31. Qing Z, Su H, Gan W, Wang D, Wu W, Wang X, Qiao Y, Yan J, Gao C, Sang N (2021) Temporal context aggregation network for temporal action proposal refinement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 485–494
32. Shou Z, Wang D, Chang SF (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
33. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos advances in neural information processing systems, vol 27
34. 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, pp 2602–2610
35. Tan J, Tang J, Wang L, Wu G (2021) Relaxed transformer decoders for direct action proposal generation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 13526–13535
36. Tian F, Gao Y, Fang Z, Fang Y, Gu J, Fujita H, Hwang JN (2021) Depth estimation using a self-supervised network based on cross-layer feature fusion and the quadtree constraint IEEE transactions on circuits and systems for video technology

37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need advances in neural information processing systems, vol 30
38. Wang L, Xiong Y, Lin D, Van Gool L (2017) Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4325–4334
39. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision. Springer, pp 20–36
40. 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
41. Wang X, Shi J, Fujita H, Zhao Y (2021) Aggregate attention module for fine-grained image classification. J Ambient Intell Humanized Comput, pp 1–11
42. Wang Y, Long M, Wang J, Yu PS (2017) Spatiotemporal pyramid network for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1529–1538
43. Wang Y, Xu Z, Wang X, Shen C, Cheng B, Shen H, Xia H (2021) End-to-end video instance segmentation with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8741–8750
44. Wu Y, Jiang X, Fang Z, Gao Y, Fujita H (2021) Multi-modal 3d object detection by 2d-guided precision anchor proposal and multi-layer fusion. Appl Soft Comput 107405:108
45. Xia K, Wang L, Zhou S, Hua G, Tang W (2022) Dual relation network for temporal action localization. Pattern Recogn 108725:129
46. Xiong Y, Wang L, Wang Z, Zhang B, Song H, Li W, Lin D, Qiao Y, Van Gool L, Tang X (2016) Cuhk & ethz & siat submission to activitynet challenge 2016. arXiv:1608.00797
47. Xu J, Chen G, Zhou N, Zheng WS, Lu J (2022) Probabilistic temporal modeling for unintentional action localization. IEEE Trans Image Process 31:3081–3094
48. Xu M, Zhao C, Rojas DS, Thabet A, Ghanem B (2020) G-tad: sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10156–10165
49. Yan B, Peng H, Fu J, Wang D, Lu H (2021) Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10448–10457
50. Yang L, Peng H, Zhang D, Fu J, Han J (2020) Revisiting anchor mechanisms for temporal action localization. IEEE Trans Image Process 29:8535–8548
51. Yao G, Lei T, Zhong J, Jiang P (2019) Learning multi-temporal-scale deep information for action recognition. Appl Intell 49(6):2017–2029
52. Yao Y, Jiang X, Fujita H, Fang Z (2022) A sparse graph wavelet convolution neural network for video-based person re-identification. Pattern Recogn 129:108708
53. Zaheer M, Guruganesh G, Dubey KA, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L et al (2020) Big bird: transformers for longer sequences. Adv Neural Inf Process Syst 33:17283–17297
54. Zeng R, Huang W, Tan M, Rong Y, Zhao P, Huang J, Gan C (2019) Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7094–7103
55. Zeng R, Huang W, Tan M, Rong Y, Zhao P, Huang J, Gan C (2021) Graph convolutional module for temporal action localization in videos. IEEE Trans Pattern Anal Mach Intell
56. Zhai Y, Wang L, Tang W, Zhang Q, Yuan J, Hua G (2020) Two-stream consensus network for weakly-supervised temporal action localization. In: European conference on computer vision. Springer, pp 37–54
57. Zhao P, Xie L, Ju C, Zhang Y, Wang Y, Tian Q (2020) Bottom-up temporal action localization with mutual regularization. In: European conference on computer vision. Springer, pp 539–555
58. 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
59. Zhao Y, Xiong Y, Wang L, Wu Z, Tao X, Lin D (2020) Temporal action detection with structured segment networks. Int J Comput Vis 128:74–95
60. Zhao Y, Zhang H, Gao Z, Guan W, Nie J, Liu A, Wang M, Chen S (2022) A temporal-aware relation and attention network for temporal action localization. IEEE Trans Image Process
61. Zhou Y, Wang R, Li H, Kung SY (2020) Temporal action localization using long short-term dependency. IEEE Trans Multimedia 23:4363–4375
62. Zhu K, Jiang X, Fang Z, Gao Y, Fujita H, Hwang JN (2021) Photometric transfer for direct visual odometry. Knowl-Based Syst 106671:213
63. 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/CVF international conference on computer vision, pp 13516–13525

Yu Yang received the B.S. degree in control science and engineering at the School of Automation from China University of Geosciences, Hubei, China, in 2021. He is currently pursuing the M.S. degree with the College of Control Science and Engineering, Zhejiang University, Zhejiang, China. His research interests include action recognition, LiDAR segmentation, computer vision, and deep learning.

Mengmeng Wang received the B.S. and M.S. degrees in control science and engineering from Zhejiang University, Zhejiang, China, in 2015 and 2018, respectively, where she is currently pursuing the Ph.D. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering. Her research interests include visual tracking, action recognition, computer vision, and deep learning.

Jianbiao Mei received the B.S. degree in control science and engineering from Zhejiang University, Zhejiang, China, in 2021, where he is currently pursuing the M.S. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering. His research interests include video object segmentation, computer vision, and deep learning.

Yong Liu received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Zhejiang, China, in 2001 and 2007, respectively. He is currently a Professor with the Institute of Cyber-Systems and Control, Zhejiang University. His main research interests include robot perception and vision, deep learning, big data analysis, multi-sensor fusion, machine learning, computer vision, information fusion, and robotics.