Temporal Action Localization (TAL) is a significant and challenging task that searches for subtle human activities in an untrimmed video. To extract snippet-level video features, existing TAL methods commonly use video encoders pre-trained on short-video classification datasets. However, the snippet-level features can incur ambiguity between consecutive frames due to short and poor temporal information, disrupting the precise prediction of action instances. Several methods incorporating temporal relations have been proposed to mitigate this problem; however, they still suffer from poor video features. To address this issue, we propose a novel temporal action localization framework called an Action-aware Masking Network (AMNet). Our method simultaneously refines video features using action-aware attention and considers inherent temporal relations using self-attention and cross-attention mechanisms. First, we present an Action Masking Encoder (AME) that generates an action-aware mask to represent positive characteristics, which is then used to refine snippet-level features to be more salient around actions. Second, we design a Group Attention Module (GAM), which models relations of temporal information and exchanges mutual information by dividing the features into two groups, i.e., long and short-groups. Extensive experiments and ablation studies on two primary benchmark datasets demonstrate the effectiveness of AMNet, and our method achieves state-of-the-art performances on THUMOS-14 and ActivityNet1.3.

1. Introduction

Temporal Action Localization (TAL) is a core task in video understanding. TAL has attracted attention recently, which can be extended to various video-related studies [45], e.g., video retrieval [16, 11], video surveillance [46, 7], and video summarization [43, 10]. Given an untrimmed video, TAL aims to predict start time, end time, and category of actions. It is a challenging task because classification and localization are conducted simultaneously to find complex and vague action instances in the long untrimmed video.

In TAL, various suitable methods have recently been proposed, with most approaches [35, 39, 14, 48, 17] commonly relying on pre-trained video encoders. Specifically, an untrimmed video is split into snippets and features are extracted from every snippet. Then, with the extracted features, the proposed action detection model is used to predict action boundaries and categories.

Existing methods are diverse and have remarkable performances but do not fully utilize inherent semantic information due to limited feature representation. In particular, the video encoders, pre-trained for video-level classification, are not optimized in TAL, and so, the extracted features with snippet-level videos do not provide sufficient contextual information. This is because the snippet-level video contains around 8 to 32 frames. Assuming that the video is 30 fps, it is approximately 0.27 to 1.07 seconds. This limitation causes ambiguity between consecutive frames resulting in the TAL model not being able to distinguish clearly between frames of action and background, which can hinder subsequent detection and classification processes. This ambiguity not only interrupts the precise prediction of action boundaries but also results in inconsistency between classification and localization. Even if the predicted temporal action boundary is exact, inaccurate classification scores can negatively affect the detection performance by non-maximum suppression (NMS) [4].

In this paper, we propose an Action-aware Masking Network (AMNet) to address the scene ambiguity through action-aware attention and self-attention. We first characterize original snippet-level features with positive and negative components based on the ground truth in the training...
stage. Here, positive and negative parts denote the areas of actions and background, respectively. We then train an action-aware attention mask to represent a positive component by maintaining a considerable embedded distance from the negative. Using this mask, we refine the original representation to make it more pronounced in the action areas, considering inherent semantic information. Furthermore, to utilize the refined feature, we divide the feature into multi-scale features and apply a self and cross-attention mechanism.

Our proposed framework consists of three main components: (i) an Action Masking Encoder (AME), (ii) a Group Attention Module (GAM), and (iii) prediction heads such as class, boundary, and matching score heads. The AME generates the action-aware mask from the video feature, masking it as a residual-alike approach. The masked feature benefits temporal action information, maintaining existing feature information. The GAM contains a feature pyramid network that generates multi-scale features to cover action detection of various lengths. Existing methods [25, 42] process each multi-scale feature independently. However, this approach cannot fully utilize a multi-scale structure with different inherent temporal information. As the feature with a long temporal dimension tends to focus on the local context and the feature with a short temporal dimension tends to focus on the global context, we combine the multi-scale features into two groups, i.e., long and short groups. We then conduct cross-attention between the two groups to compensate for the lack of knowledge. Our prediction heads consist of class, boundary, and matching score heads. The matching score head generates matching scores, which are further multiplied by the classification scores.

Our proposed AMNet demonstrates effectiveness by conducting extensive experiments on two benchmark datasets: THUMOS-14 [18] and ActivityNet1.3 [5]. As a result, we achieve state-of-the-art performance, and our contributions can be summarized as follows:

- We propose an AMNet, in which the AME generates an action-aware mask that refines the snippet-level video feature by applying action-aware attention to address the scene ambiguity. It emphasizes the action area of the feature by masking the original video feature.

- We design a GAM, which models the inherent temporal relation by combining multi-scale features into two groups and applying cross-attention.

- We conduct extensive experiments, and our method outperforms other state-of-the-art methods on two primary datasets, i.e., THUMOS-14 and ActivityNet1.3.

2. Related Work

2.1. Action Recognition

Action recognition [1, 37] has been actively studied for a long time as an area of pattern recognition [36, 23, 22, 24, 15] and a fundamental task for TAL. The traditional action recognition methods can be divided into skeleton-based methods (Shift-GCN [9]), and video-based methods (TSN [41] and I3D [6]). The I3D model, which is a two-stream inflated 3D convolutional network utilizing RGB and optical flow, is most prevalent in TAL. The I3D increases the receptive fields of 2D CNN by inflating the convolution filters and kernel sizes of pooling, thereby considering temporal dimensions. We adopted the I3D model pre-trained on the Kinetics dataset [21] because of its superior ability for action recognition. However, the snippet-level video features extracted by the video encoder can have limited temporal information because of the short-term snippet-level videos. Our proposed method focuses on mitigating this problem.

2.2. Temporal Action Localization

Unlike action recognition, the datasets for TAL are untrimmed long videos. Furthermore, TAL conducts two tasks simultaneously, namely classification and localization of actions. Overall TAL process can be divided into three steps: (i) feature extraction, (ii) prediction using the TAL model, and (iii) post-processing using Soft-NMS [4], as shown in Fig. 1. Most TAL methods [39, 20, 44] utilize the pre-trained action recognition model as the backbone architecture to extract video features. With these extracted features, the TAL methods focus on the prediction stage. However, we argue that offline snippet-level features can be sub-optimal for localization actions because of insufficient temporal knowledge. To address this issue, we refine the snippet-level video features by conducting action-aware attention with an action-aware mask generated by the proposed AME.
3. Proposed Method

In this section, we present a novel TAL framework called an Action-aware Masking Network (AMNet), which consists of three main components: an Action Masking Encoder (AME), a Group Attention Module (GAM), and prediction heads. Specifically, we refine video features with an action-aware mask generated through AME and model each relation of multi-scale features by grouping through GAM. In training, our method is asynchronously processed in two steps; therefore we first explain (i) mask representation learning. We then introduce (ii) action detector learning. The overall pipeline of our method is shown in Fig. 2.

3.1. Problem Settings and Feature Extraction

Given an untrimmed video, TAL aims to predict the actions’ start time, end time, and confidence score. As a first step, we extract the video feature $F$ for each snippet-level video, which contains a few frames (e.g., 16 frames), using the pre-trained video encoder [6]. The extracted video feature can be denoted as $F \in \mathbb{R}^{T \times C}$, where $T$ and $C$ are temporal dimension and channels.

3.2. Mask Representation Learning

In mask representation learning, we train the AME, generating an action-aware mask to refine the video feature $F$ through action-aware attention. Specifically, according to the ground truth, we divide the video feature into positive (Action) and negative (Background) components as shown in Fig. 2. Then, we collect and concatenate the corresponding snippet-level features along the temporal dimension. The positive $F_{pos} \in \mathbb{R}^{T_p \times C}$ and negative features
conducting action-aware attention, we can observe that the positive and negative features must have orthogonal properties. Furthermore, the mask must be able to represent each attention. To this end, we adopt a triplet loss [38] widely used for feature representation learning or clustering. To briefly explain it, we set the mask $F_{\text{mask}}$ to anchor and find the Euclidean distance of the embedded anchor, positive and negative, as follows:

$$d_{\text{pos}} = \left\| F_{\text{mask}} - \hat{F}_{\text{pos}} \right\|^2_2,$$
$$d_{\text{neg}} = \left\| F_{\text{mask}} - \hat{F}_{\text{neg}} \right\|^2_2.$$  

(2)

Here, we intend to minimize $d_{\text{pos}}$ and maximize $d_{\text{neg}}$. So, the triplet loss $L_{\text{trip}}$ can be formulated as:

$$L_{\text{trip}} = [d_{\text{pos}} - d_{\text{neg}} + \alpha]_+,$$

(3)

where $\alpha$ denotes a margin enforced between positive and negative pairs. With this loss, we can obtain the action-aware mask with AME that encodes the feature salient to the positive one.

### 3.3. Action Detector Learning

In action detector learning, we start to train our AM-Net in earnest. First, we introduce a detailed refinement of the video feature process using AME. Next, we present the structure of the GAM for modeling inherent temporal relations between long and short groups. Finally, we explain about three prediction heads: (i) class head, (ii) boundary head, and (iii) matching score head. The details are explained below.

**Refinement of Video Feature** To obtain the optimal feature for TAL that has salient values around the action area, we first generate an action-aware mask using the AME. Next, we obtain a masked feature by action-aware attention, conducting the residual-alike operation as follows:

$$F_{\text{mask}} = \text{AME}(F),$$
$$\hat{F} = F + F_{\text{mask}},$$

(4)

where $F \in \mathbb{R}^{T \times C}$ and $F_{\text{mask}} \in \mathbb{R}^{T \times C}$ denote the video feature and the action-aware mask, respectively. After conducting action-aware attention, we can observe that the masked feature is refined to be salient around the action area, as shown in Fig. 3. Afterward, the masked feature $\hat{F} \in \mathbb{R}^{T \times C}$ is used for input of GAM.

**Group Attention Module (GAM)** To fully utilize the inherent semantic knowledge of the masked feature $\hat{F}$, we build a GAM that models the temporal relations of each time step, as shown in Fig. 4. To obtain temporal action boundaries of various lengths, the masked feature $\hat{F}$ is divided into $K$ multi-scale features $\{F^i_m \in \mathbb{R}^{T \times C}\}_{i=1}^K$ by the feature pyramid network, which consists of 1D CNNs. Each multi-scale feature has different temporal dimensions, reduced by half, respectively. Afterward, we conduct self-attention on each multi-scale feature $(F^1_m, F^2_m, \cdots, F^K_m)$ to model the relation between each temporal location. First, the multi-scale features are projected into query $Q$, key $K$, and value $V$, respectively as follows:

$$\begin{align*}
Q_i &= W^q_i \cdot F^i_m, \\
K_i &= W^k_i \cdot F^i_m, \\
V_i &= W^v_i \cdot F^i_m 
\end{align*}$$

(5)

where $W$ denotes a learnable weight that projects the feature into query, key, and value. With these projected features, we conduct a self-attention operation, which is formulated as follows:

$$\text{att}_i = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{D}}) V_i,$$

(6)

where $D$ denotes the channel of each attention head. The channel $D$ is calculated as $\frac{C}{N_h}$ where $N_h$ is the number of attention heads.

After conducting the self-attention operation, we combine the multi-scale features into two groups: long and short groups, based on the length of the temporal dimension as follows:
where $\lfloor \cdot \rfloor$ and $g(\cdot)$ denote temporal-wise concatenation and self-attention, respectively. Note that the features with various temporal dimensions benefit from generating proposals of various lengths. Specifically, the feature with a longer temporal dimension, which focuses on local context, tends to generate relatively short action boundaries. In contrast, the feature with a shorter temporal dimension, which focuses on the global context, tends to generate relatively long action boundaries. It is because the predicted absolute distance values of start and end from specific time steps have a lower percentage in the long temporal dimension than in the short temporal dimension, and vice versa. So, we conduct cross-attention between two groups (long and short) to compensate for the lack of semantic knowledge as follows:

$$
G_{\text{short}} = [g(F^1_m), \ldots, g(F^K_m)],
G_{\text{long}} = [g(F^K_m^1), \ldots, g(F^K_m)],
$$

where $\lfloor \cdot \rfloor$ and $g(\cdot)$ denote temporal-wise concatenation and self-attention, respectively. Note that the features with various temporal dimensions benefit from generating proposals of various lengths. Specifically, the feature with a longer temporal dimension, which focuses on local context, tends to generate relatively short action boundaries. In contrast, the feature with a shorter temporal dimension, which focuses on the global context, tends to generate relatively long action boundaries. It is because the predicted absolute distance values of start and end from specific time steps have a lower percentage in the long temporal dimension than in the short temporal dimension, and vice versa. So, we conduct cross-attention between two groups (long and short) to compensate for the lack of semantic knowledge as follows:

$$
Q_S = W_{cq} \cdot G_{\text{short}},
Q_L = W_{cq} \cdot G_{\text{long}},
K_S = W_{ck} \cdot G_{\text{short}},
K_L = W_{ck} \cdot G_{\text{long}},
V_S = W_{cv} \cdot G_{\text{short}},
V_L = W_{cv} \cdot G_{\text{long}},
G_{L\rightarrow S} = \text{MLP}(\epsilon(M\text{CA}(K_S, V_S, Q_L))),
G_{S\rightarrow L} = \text{MLP}(\epsilon(M\text{CA}(K_L, V_L, Q_S))),
$$

where $\epsilon$, MLP and MCA denote layer normalization, multilayer perceptron, and multi-head cross-attention, respectively. We then reshape each $G_{L\rightarrow S}$ and $G_{S\rightarrow L}$ as the shape of the original multi-scale features denoted as $\{\hat{F}_i \in \mathbb{R}^{T_i \times C}, i = 1, \ldots, K\}$ before predicting the final outputs.

**Prediction Heads** General TAL methods predict two outputs: temporal boundary and action category. However, these methods unfortunately often neglect the inconsistency between classification and localization derived from scene ambiguity, which is one of the main factors that cause performance degradation. Therefore, we add auxiliary output, the matching score, to make the confidence scores robust against incorrect suppression by Soft-NMS [4] in inference time.

Our prediction heads (i.e., class, boundary, and matching score heads) are composed with 1D convolutional layers. They use the main block of the same structure as follows:

$$
\text{Block}(x) = \{\sigma(\epsilon(\text{Conv1d}(x)))\}_{i=1}^{K},
$$

where $\sigma$, $\epsilon$, and $K$ denote an activation function, layer normalization, and the number of layers, respectively. The final outputs, such as the temporal boundaries, confidence scores, and matching score, are generated as follows:

$$
\hat{y}_i = \text{FC}(\text{Block}(\hat{F}_i)),
\hat{B}_i = \sigma(\text{FC}(\text{Block}(\hat{F}_i)) \times \omega_B),
\hat{m}_i = \text{FC}(\text{Block}(\hat{F}_i)) \times \omega_M,
$$

where $\hat{y}_i \in \mathbb{R}^{T_i \times N_C}$, $\hat{B}_i \in \mathbb{R}^{T_i \times 2}$, and $\hat{m}_i \in \mathbb{R}^{T_i \times 1}$ denote the predicted confidence score with $N_C$ classes, temporal boundary, and matching score, respectively. In addition, we adjust the scales of boundaries and matching scores through the learnable weights $\omega_B$ and $\omega_M$, respectively.

**3.4. Loss Function**

In this section, we introduce the loss functions of our proposed method. As mentioned above, we train our model in two phases: (i) mask representation learning and (ii) action detector learning. The triplet loss $L_{\text{trip}}$ in the mask representation learning first processes back-propagation. And then, we conduct back-propagation of the losses in the action detector learning.

The losses of the action detector consist of class $L_{\text{cls}}$, boundary $L_{\text{reg}}$, and matching score $L_{\text{mat}}$ losses. We adopt a focal loss [28] for classification, which alleviates the class imbalance problem. Also, we use an IoU loss for the boundary regression, which calculates the percentage of overlapping the predicted boundaries $\hat{B} = (\hat{t}^s, \hat{t}^e)$ and ground truths, where $\hat{t}^s$ and $\hat{t}^e$ denote the action’s start time and end time, respectively. Furthermore, we use the mean squared error between the matching score $\hat{m}$ and the IoU value of the predicted boundary for the matching loss. Here, we normalize the matching score using a hyperbolic tangent function, which enriches the output range and slightly increases the performance than using a sigmoid function, as shown in Tab. 4. These losses can be formulated as follows:

$$
L_{\text{cls}} = \sum_k (\text{FL}(\hat{y}_k, y_k)),
L_{\text{reg}} = \sum_k (1 - \text{IoU}(\hat{B}_k, B_k)),
L_{\text{mat}} = \sum_k (\text{tanh}(\hat{m}_k) - \text{IoU}(\hat{B}_k, B_k))^2,
$$

where $y$, $\hat{y}$, and FL denote the ground truth of class, predicted confidence score, and the focal loss, respectively.

The total loss can be formulated as:

$$
L = L_{\text{cls}} + \lambda_1 (L_{\text{reg}} + L_{\text{mat}}) + \lambda_2 L_{\text{trip}},
$$

where $\lambda_1$ and $\lambda_2$ denote the weights balancing between the losses.

**3.5. Inference**

Given an untrimmed video $X$, our method outputs the distances from each time steps $\{(d^s_i, d^e_i)\}_{i=1}^T$, confidence score $\hat{y}$, and matching score $\hat{m}$, where $i$ denotes the time steps. From the distances, we calculate the boundaries $\hat{B}_i = (\hat{t}^s_i, \hat{t}^e_i)$ as follows:

$$
\hat{t}^s_i = i - d^s_i,
\hat{t}^e_i = i + d^e_i.
$$
Table 1. Comparison of our method with other state-of-the-art methods on THUMOS14 and ActivityNet datasets. The results are measured by mAP (%) at different tIoU thresholds. The second column (Feature) denotes each method’s video encoder.

<table>
<thead>
<tr>
<th>Method</th>
<th>Feature</th>
<th>THUMOS14</th>
<th></th>
<th>ActivityNet1.3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>BSN (ECCV’18) [27]</td>
<td>TSN [41]</td>
<td>53.5</td>
<td>45.0</td>
<td>36.9</td>
<td>28.4</td>
</tr>
<tr>
<td>BMN (ICCV ’19) [26]</td>
<td>TSN [41]</td>
<td>56.0</td>
<td>47.4</td>
<td>38.8</td>
<td>29.7</td>
</tr>
<tr>
<td>G-TAD (CVPR’20) [44]</td>
<td>TSN [41]</td>
<td>54.5</td>
<td>47.6</td>
<td>40.3</td>
<td>30.8</td>
</tr>
<tr>
<td>TCA-Net (CVPR’21) [34]</td>
<td>TSN [41]</td>
<td>60.6</td>
<td>53.2</td>
<td>44.6</td>
<td>36.8</td>
</tr>
<tr>
<td>RTD-Net (ICCV’21) [39]</td>
<td>I3D [6]</td>
<td>68.3</td>
<td>62.3</td>
<td>51.9</td>
<td>38.8</td>
</tr>
<tr>
<td>ContextLoc (ICCV’21) [48]</td>
<td>I3D [6]</td>
<td>68.3</td>
<td>63.8</td>
<td>54.3</td>
<td>41.8</td>
</tr>
<tr>
<td>AFSD (CVPR’21) [25]</td>
<td>I3D [6]</td>
<td>67.3</td>
<td>62.4</td>
<td>55.5</td>
<td>43.7</td>
</tr>
<tr>
<td>MUSES (CVPR’21) [30]</td>
<td>I3D [6]</td>
<td>68.9</td>
<td>64.0</td>
<td>56.9</td>
<td>46.3</td>
</tr>
<tr>
<td>DCAN (AAAI’22) [8]</td>
<td>TSN [41]</td>
<td>68.2</td>
<td>62.7</td>
<td>54.1</td>
<td>43.9</td>
</tr>
<tr>
<td>Zhu et al. (AAAI’22) [49]</td>
<td>I3D [6]</td>
<td>72.1</td>
<td>65.9</td>
<td>57.0</td>
<td>44.2</td>
</tr>
<tr>
<td>Liu et al. (CVPR’22) [29]</td>
<td>SlowFast [13]</td>
<td>69.4</td>
<td>64.3</td>
<td>56.0</td>
<td>46.4</td>
</tr>
</tbody>
</table>

We normalize the confidence and matching scores using the sigmoid and hyperbolic tangent functions in the same manner as training. Then, we obtain a refined confidence score by multiplying each other as follows:

\[
\bar{y} = \text{sigmoid}(\hat{y}) \cdot \tanh(\hat{m}).
\] (14)

Finally, we can obtain the final outputs after conducting the soft-NMS [4] to suppress redundant proposals based on the refined confidence score.

4. Experiments

In this section, we provide extensive experiments on two primary datasets: THUMOS14 [18] and ActivityNet1.3 [5]. First, we introduce the two datasets, implementation details, and evaluation metrics used for our experiments. Next, we compare our method with previous state-of-the-art methods, and our overall results show high precision in localization and classification. Furthermore, we conduct various ablation studies to verify the effectiveness of our method. Finally, we provide an error profiling [2] that allows us to analyze our result’s false positive ratios.

4.1. Datasets

In this section, we introduce two primary datasets used for our experiments:

**THUMOS14** [18] contains 413 untrimmed videos with 20 action classes and temporal annotations. According to the public regulation, we split them into 200 videos for training and 213 videos for testing.

**ActivityNet1.3** [5] contains 19,994 untrimmed videos with 200 action classes and temporal annotations, which is much larger than THUMOS14. According to the setting of prior works [27, 26, 44], we split the videos into 10,024 videos for training, 4,926 videos for validation, and 5,044 videos for testing by a 2:1:1 ratio.

4.2. Implementation Details

For the THUMOS14 dataset, we train our model for 45 epochs using AdamW [33] optimizer. The batch size is 4 and weight decay is set to \(5 \times 10^{-2}\). We set a learning rate to \(10^{-4}\) and adopt a cosine annealing [32] manner. We use the I3D [6] model, pre-trained on Kinetics dataset [21], to extract the video features from the video using a sliding window covering 16 frames with 4 strides. The loss weight parameters \(\lambda_1\) and \(\lambda_2\) are set to 1, which performed best in the ablation study in Tab. 5.

For the ActivityNet1.3 dataset, we train our model for 10 epochs using AdamW optimizer. The batch size is 16 and weight decay is set to \(5 \times 10^{-2}\). We set a learning rate to \(10^{-5}\) and adopt a cosine annealing manner. We use the I3D model, pre-trained on Kinetics dataset, to extract the video features from the video using a sliding window covering 16 frames without overlapping, \(i.e.,\) 16 strides. The loss weight parameters \(\lambda_1\) and \(\lambda_2\) are set to 1 as same as THUMOS14 settings. Furthermore, following [27, 44], we utilize the score fusion manner for reliable results. The video classification scores from [47] are multiplied by the confidence score in the inference time.

4.3. Evaluation Metrics

In our experiments, we use mean Average Precision (mAP) to evaluate TAL performance, which is the mean value for the average precision of each action class. Following traditional practice, the temporal Intersection over Union (tIoU) thresholds are set to \([0.3:0.1:0.7]\) for THUMOS14 and \([0.5:0.05:0.95]\) for ActivityNet1.3.

4.4. Main Results

In this section, we demonstrate the effectiveness of our method by comparing it with other state-of-the-art methods.
on THUMOS14 and ActivityNet1.3, as shown in Tab. 1.

**THUMOS14** We compare our method with other state-of-the-art methods on THUMOS14 in Tab. 1. Our method noticeably achieves the superior mAP at all thresholds, reaching 63.3%. In particular, our method surpasses the Zhu *et al.* [49] method by +4.6% mAP@0.3 absolute improvement, reaching 76.7%. Furthermore, our method outperforms the previous state-of-the-art method (Liu *et al.* [29]) by +7.8% mAP@0.7 absolute improvement.

**ActivityNet1.3** We compare our method with other state-of-the-art methods on ActivityNet1.3 in Tab. 1. At tIoU=0.75, we achieve the highest mAP, which surpasses the TCA-Net [34] method by 1.0% absolute improvement, reaching 37.7%. Furthermore, although our method does not achieve the highest mAP@0.5 and mAP@0.95, we outperform other methods with a 0.9% gap at mAP@Avg. We guess two reasons for weaker performance improvement than on THUMOS14: First, it is more challenging to classify because ActivityNet1.3 has more action categories (200 classes) than THUMOS14 (20 classes). Second, because the temporal locations of ground truths are not diverse, the action detector is overfitted on the biased situations.

**4.5. Ablation Study**

**Effectiveness of Proposed Modules** We evaluate the effectiveness of our key modules, such as AME, GAM, and the matching score head (MS), as shown in Tab. 2. We adopt the anchor-free method [25] as the baseline model (1st row), which is improved through various training techniques such as cosine annealing and label smoothing with optimal parameter choices. In 2nd row, the result shows that the matching score head mitigates the inconsistency problem by redefining the confidence score, improving +2.3% mAP@Avg compared to the baseline. Furthermore, in 3rd and 4th rows, we can observe that our key modules AME and GAM con-

**Table 2. Ablation study of the proposed modules such as AME, GAM, and matching score head (MS) on THUMOS14.**

<table>
<thead>
<tr>
<th>AME</th>
<th>GAM</th>
<th>MS</th>
<th>THUMOS14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
</tr>
<tr>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>77.2</td>
</tr>
<tr>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>73.9</td>
</tr>
<tr>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>75.5</td>
</tr>
<tr>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>76.4</td>
</tr>
</tbody>
</table>

Table 3. Ablation study of different attention on THUMOS14. The baseline model is the same as 2nd row in Tab. 2, which consists of the matching score head.

<table>
<thead>
<tr>
<th>Confidence Score</th>
<th>0.3</th>
<th>THUMOS14</th>
</tr>
</thead>
<tbody>
<tr>
<td>sigmoid(ŷ)</td>
<td>75.9</td>
<td>63.8</td>
</tr>
<tr>
<td>sigmoid(ŷ) · sigmoid(ω)</td>
<td>76.5 (+0.6)</td>
<td>65.4 (+1.6)</td>
</tr>
<tr>
<td>sigmoid(ŷ) · tanh(ω)</td>
<td>76.7 (+0.8)</td>
<td>66.8 (+3.0)</td>
</tr>
</tbody>
</table>

Table 4. Ablation study of different designs of confidence score on THUMOS14.

<table>
<thead>
<tr>
<th>λ1</th>
<th>λ2</th>
<th>THUMOS14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.3</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>76.7</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>76.0</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>76.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>76.7</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>76.0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>75.8</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>76.5</td>
</tr>
</tbody>
</table>

Table 5. Ablation study of the balanced weight between the different losses on THUMOS14.
considerably improve the performance with +3.5% and +5.1% mAP@Avg compared to the baseline. Finally, our complete model improves the performance by +8.9% mAP@Avg compared to the baseline. We also conduct the ablation study of each attention effect in Tab. 3, where the baseline (1st row) is the same as the baseline added the corresponding attention. The results show that group-based attention carries the highest gain by +4.8 mAP@Avg compared to the baseline. These ablation studies demonstrate that the AME and GAM contribute significantly to performance improvement. Additionally, we provide the detailed comparison (Fig. 5) of per-class between the baseline (1st row in Tab. 2), the baseline with GAM and MS (3rd row in Tab. 2), and our complete model.

Refinement of Confidence Score To verify the effectiveness of refining the confidence score with matching score, we conduct an ablation study by changing the design of confidence score, as shown in Tab. 4. 1st row denotes the result when our model infers using the vanilla confidence score trained without the matching score head. 2nd and 3rd rows denote the refined confidence scores by different matching scores. The results show that the hyperbolic tangent function slightly improves performance than the sigmoid function. We conjecture it is because the hyperbolic tangent function widens the matching score range.

Matching Loss To choose the suitable loss of the matching score, we experiment with the different designs of the matching losses $L_{mat}$ on THUMOS14 dataset, as shown in Fig. 6. In the case of binary cross entropy (BCE), we replace the hyperbolic tangent function (eq. 11) with the sigmoid function, as the input of BCE must be positive values. The results show that L2 loss is the most stable and robust for predicting tIoU values of temporal boundaries.

Balancing Weights between Losses To find the optimal balancing weights, we conduct a grid search on THUMOS14 dataset, as shown in Tab. 5. First, we set the two hyper-parameters: $\lambda_1$ for regression losses and $\lambda_2$ for the triplet loss, considering the weight for classification loss to 1, and we set the weight range to [0.5:0.5:2]. As a result, we can observe that the setting when all weights are equivalent yields the best performance.

Errors of Our Result To analyze the limitations of our model, we provide the false positive error chart [2] of our detection results. The experiment results are reported at the fixed 0.5 tIoU threshold on THUMOS14 dataset. As shown in Fig. 7, we can observe that the impact of localization and background errors is significant. We expect a more precise regression loss design to mitigate them in further works.

5. Conclusion In this paper, we propose a novel temporal action localization framework called AMNet, to address the ambiguity between consecutive frames caused by poor temporal information of video features. In particular, we present an AME to represent semantic action features and explicitly apply action-aware attention to video features extracted from a pre-trained video encoder. Furthermore, we propose a GAM to model temporal semantic knowledge by grouping multi-scale features. The extensive experimental results on THUMOS14 and ActivityNet1.3 demonstrated that our AMNet has high fidelity of localization and classification and can therefore achieve state-of-the-art performance.

Acknowledgement This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program(Korea University), No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation).

Figure 6. The ablation study of different matching loss designs on THUMOS14, measured by mAP (%) at different tIoU thresholds.

Figure 7. Error chart of our detection result, drawn up using DETAD [2]. There are error rates of 5 types on top-10G predictions, where G denotes the number of ground truths. Detailed instructions about the chart are in DETAD [2].
References


