Learning Generalized Representations for Open-Set Temporal Action Localization

Junshan Hu  
University of Science and Technology of China

Liansheng Zhuang*  
University of Science and Technology of China

Weisong Dong  
University of Science and Technology of China

Shiming Ge  
Institute of Information Engineering, Chinese Academy of Sciences

Shafei Wang  
Peng Cheng Laboratory  
Shenzhen, Guangdong, China

ABSTRACT

Open-set Temporal Action Localization (OSTAL) is a critical and challenging task that aims to recognize and temporally localize human actions in untrimmed videos in open word scenarios. The main challenge in this task is the knowledge transfer from known actions to unknown actions. However, existing methods utilize limited training data and overparameterized deep neural network, which have poor generalization. This paper proposes a novel Generalized OSTAL model (namely GOTAL) to learn generalized representations of actions. GOTAL utilizes a Transformer network to model actions and an open-set detection head to perform action localization and recognition. Benefitting from Transformer’s temporal modeling capabilities, GOTAL facilitates the extraction of human motion information from videos to mitigate the effects of irrelevant background data. Furthermore, a sharpness minimization algorithm is used to learn the network parameters of GOTAL, which facilitates the convergence of network parameters towards flatter minima by simultaneously minimizing the training loss value and sharpness of the loss plane. The collaboration of the above components significantly enhances the generalization of the representation. Experimental results demonstrate that GOTAL achieves the state-of-the-art performance on THUMOS14 and ActivityNet1.3 benchmarks, confirming the effectiveness of our proposed method.

CCS CONCEPTS

• Computing methodologies → Activity recognition and understanding.

KEYWORDS

Video understanding, open-set temporal action localization, Transformer, generalization

1 INTRODUCTION

Temporal Action Localization (TAL), aiming to temporally recognize and locate human actions in untrimmed videos, is a challenging video understanding problem. With the remarkable advances in video understanding [9, 19] and object detection [7, 8, 33], TAL has been made significant breakthroughs. However, previous TAL methods tend to fall short in reproducing their excellent performance on the test set when applied in practical situations. This is because most methods use the closed-set assumption that the test set has only a predefined and limited number of categories.
However, in practice, unknown human actions are inevitable to appear in an open world. As a result, unknown actions are often incorrectly classified as known actions, which increases the false positive rate.

To relax the above closed-set condition, the Open-Set Temporal Action Localization (OSTAL) [3] considers a realistic scenario where test videos might include novel actions that were not present during training. The aim of OSTAL is to not only temporally localize and recognize the known actions but also reject the localized unknown actions. The category number of known actions that are annotated in standard datasets like THUMOS14 [30] and ActivityNet-1.3 [24] are often very low (20 and 200 respectively) when compared to the infinite number of actions that are present in the open world. Recognizing an unknown action as unknown requires strong generalization. As shown in Fig.1, a model with weak generalization may overfit the background, tending to recognize unknown actions as known actions. Presently, prevailing models typically utilize deep neural networks with numerous parameters, and learn the parameters by minimizing the empirical error on the training set. Although various techniques (such as batch normalization [26] and Dropout [50]) are employed to prevent overfitting, deep learning models demonstrate poor generalization when operating in open-world scenarios. Increasing the number of training samples can improve the generalization. Nonetheless, collecting video data and manually annotating each frame of the video is time-consuming and labor-intensive.

In order to improve the generalization of open-set temporal action detection models, this paper proposes a novel Generalized OSTAL model (namely GOTAL), a one-stage framework for the OSTAL task. Our framework is based on a Transformer network and a open-set detection head. The former is used to model temporal actions, while the latter performs action localization and recognition. Benefitting from the powerful temporal modeling capabilities of the Transformer, GOTAL extracts human motion information from videos to eliminate irrelevant information such as background. Note here that, though the Transformer network has been employed in closed-set scenarios (such as ActionFormer [59]), its application in open-set scenarios has not been explored. As shown in our experiments, ActionFormer does not perform well in open-set scenarios. Since GOTAL is a heavily overparameterized model, the value of the training loss provides few guarantees on model generalization ability. Motivated by prior work connecting the geometry of the loss landscape and generalization, GOTAL adopts the Sharpness-Aware Minimization method (SAM) [20] to learn the network parameters by simultaneously minimizing both the loss value and the loss sharpness. SAM causes the parameters to converge towards flatter minima and helps GOTAL achieve a better generalization ability. Extensive experiments show that our method outperforms state-of-the-art methods in realistic open-set scenarios.

In summary, our main contributions are as follows:

- We propose a novel one-stage framework (namely GOTAL) for OSTAL tasks to improve the performance in the realistic scenario by enhancing the generalization of the model.
- We present the first application of Sharpness-aware Minimization to the challenging OSTAL task and justify its effectiveness for improving the generalization of GOTAL.

- Experiments show that our proposed method achieves state-of-the-art open-set performance on THUMOS14 and ActivityNet1.3 benchmarks.

2 RELATED WORK

Temporal Action Localization. The objective of Temporal Action Localization (TAL) is to temporally recognize and locate human actions in untrimmed videos. The current TAL techniques can be broadly categorized into two paradigms: two-stage and one-stage approaches. In the two-stage approaches, class-agnostic temporal proposals are first generated, followed by classification and boundary refinement of each proposal. There have been several prior studies that have concentrated on action proposal generation techniques. Some of these methods include classifying anchor windows [6, 18, 25] or detecting action boundaries [22, 35, 37, 60]. More recent approaches to this problem make use of a graph representation [1, 57]. Some other researchers have incorporated both proposal generation and classification into a unified model [11, 48, 61]. One-stage methods aim to localize actions in a single shot and do not require action proposal generation. For example, Lin et al. [36] introduced the first one-stage TAL by utilizing convolutional networks. Lin et al. [34] proposed an anchor-free model. Recently, some studies have incorporated the Transformer in TAL tasks, leading to significant improvements in detection performance. For example, some works [38, 47, 52] utilize a DETR-like Transformer-based decoder to detect action. Others work utilizes a Transformer-based encoder [14, 59] to extract a representation of the video. However, most of previous method assume that all action in videos belong to pre-defined categories, making them unsuitable for application in open-world scenarios. OpenTAL [3] is the only peer-reviewed research work in the open-set temporal action localization, which combines classification uncertainty and actionness to identify unknown actions. In this paper, building on the progress made by OpenTAL, we propose improvements to the network’s generalization.

Open-Set Recognition. In contrast to closed-set learning, which assumes that only previously known classes are present during testing, open-set learning considers the presence of both known and unknown classes. Scheirer et al. [43] were the first to introduce the concept of open-set recognition (OSR). They proposed a one-vs-rest classifier based on binary SVM, which allows for the identification of unknown samples. Subsequent studies by [28, 44] further developed the open-set framework to multi-class classifier. Bendale and Boult [5] introduced a method for identifying unknown samples in the feature space of deep networks. The proposed method, called the OpenMax classifier, employs a Weibull distribution to estimate the set risk. Current generative open OSR methods [13, 16, 21, 41] employ generative adversarial networks (GANs) [23], generative causal models, or mixup augmentation techniques to generate samples of unknown categories. Some literature [40, 51, 58] approaches OSR from a reconstruction perspective by utilizing either VAE [32] or self-supervised learning. These methods identify the unknown by reconstructing the known class data representation. Recently, probabilistic and evidential deep learning methods [2, 39, 56] that estimate uncertainty have emerged as potential methods for improving OSR performance. In this paper, we aim to the open-set
temporal action localization problem which is more challenging because of localization in open-word scenario.

**Generalization of Deep Neural Network.** The success of modern deep neural networks (DNN) in achieving state-of-the-art performance on a wide range of tasks has relied on heavier overparameterization. It is essential to learn appropriate parameters to generalize beyond the training set. In order to improve the generalization of DNN, a panoply of methods for modifying the training process have been proposed, including dropout [50], batch normalization [26], data augmentation [15], etc. Although previous methods are widely used in current DNN model, the generalization is insufficient when applied to the open-word scenario. Some researches [17, 29, 31] have shown a connection between the geometry of the loss landscape and generalization, which holds the promise of facilitating novel methods [12, 20, 27] for model training that result in improved generalization. For example, Foret et al. [20] proposed Sharpness-Aware Minimization (SAM), which efficiently and effectively improves generalization ability by minimizing loss value and loss sharpness simultaneously. Enlightened by these works, this paper incorporates the current state-of-the-art generalization method into the TAL model.

3 PROPOSED METHOD

**Problem Formulation.** An untrimmed video can be depicted as a frame sequence $X = \{x_1, x_2, \ldots, x_T\}$. A convolution backbone (e.g., I3D [10], C3D [54]) is used to extract 1D temporal feature $F^0 = \{f_1, f_2, \ldots, f_T\}$ defined on discretized time steps $t = \{1, 2, \ldots, T\}$, where $T$ varies across videos. Action annotations in video $X$ consists of $N$ action instances $Y = \{y_1, y_2, \ldots, y_N\}$. Each action instance $y_i = (s_i, e_i, c_i)$ is defined by its starting time $s_i$, ending time $e_i$ and its action label $c_i$, where $s_i, e_i \in [1, T], c_i \in \{1, \ldots, C\}$ (C is the number of pre-defined categories). The goal of temporal action localization is to predict proposals with class scores, starting time and ending time $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_M\}$, which cover $Y$ as precisely as possible.

**Representation for Action Localization.** Our method exploits an anchor-free representation for the localization of action, previously described in literature [34, 59]. It classifies every moment as an action category or the background and performs a regression of the distance between that time step and the onset and offset of the action. We define the output at time $t$ as $\hat{y}_t = (p(c_t), d_{t}^0, d_{t}^1)$, where $p(c_t)$ contains $C$ values. Each value represents a binomial variable that indicates the probability of action category $c_t \in \{1, 2, \ldots, C\}$ at time $t$. Moreover, $d_{t}^0$ and $d_{t}^1$ correspond to the distance between the current time $t$ and the onset and offset of the action, respectively. Here, $d_{t}^0, d_{t}^1 > 0$. Action localization results can be directly obtained from $\hat{y}_t = (p(c_t), d_{t}^0, d_{t}^1)$ using:

$$c_t = \arg\max p(c_t), \quad s_t = t - d_{t}^0, \quad e_t = t + d_{t}^1. \quad (1)$$

**Method Overview.** In Fig. 2, we provide an overview of our proposed GOTAL. The method revolves around a one-stage temporal action detection framework incorporating a backbone network, feature pyramid, and detection head. The backbone network leverages a convolution-based deep neural network (such as I3D [10] or TwoStream [49]) to extract video features. Next, the feature pyramid is created by the temporal action encoder with Transformer, which employs a self-attention mechanism to effectively model long-term dependencies of actions. Finally, the open-set detection head is utilized on the pyramid features to locate action boundaries and identify categories. We now detail the specifics of our model.

3.1 Temporal Action Encoder with Transformer

Initially, our model encodes an input video, $X = \{x_1, x_2, \ldots, x_T\}$, into a temporal feature $F^0 = \{f_1, f_2, \ldots, f_T\}$ by using a backbone network, with $f_i \in \mathbb{R}^D$. Afterwards, a transformer encoder maps the temporal feature to the output feature pyramid $F = \{F^1, F^2, \ldots, F^L\}$.

**Backbone network.** We adopt I3D [10] as our backbone, considering its proven success in achieving high performance in action recognition and its widespread use in previous action detection methods. For the input video $X = \{x_1, x_2, \ldots, x_T\}$, the I3D network extracts the video feature for every continuous K frame as follows: $f_i = E_{\text{I3D}}(x_{i:t}, x_{i+(K-1)}, \ldots)$, where $f_i \in \mathbb{R}^D$. Prior studies suggest that optical flow leads to enhanced model performance; hence, we employ two I3Ds to independently calculate RGB features ($I3D^{\text{RGB}}$) and optical flow features ($I3D^{\text{Flow}}$). We then proceed to concatenate these features ($f_i = [I3D^{\text{RGB}}(i), I3D^{\text{Flow}}(i)]$) to obtain the output of the backbone network ($F^0 = \{f_1, f_2, \ldots, f_T\}$).

**Transformer Encoder.** The Transformer encoder employs $F^0$ as its input. The self-attention mechanism is at the heart of the Transformer. Self-attention obtains attention weights by calculating the similarity scores between itself and other features. These weights are then used to weight and sum up the corresponding features. For $F^0 \in \mathbb{R}^{T \times D}$, comprising of $T$ time steps and $D$ dimensional feature, we project it using $W_Q \in \mathbb{R}^{D \times D_q}$, $W_K \in \mathbb{R}^{D \times D_k}$, and $W_V \in \mathbb{R}^{D \times D_v}$ to extract the feature representations of $Q$, $K$, and $V$, known as the query, key, and value respectively, while satisfying $D_k = D_q$. The $Q$, $K$, and $V$ are computed by:

$$Q = F^0 W_Q, \quad K = F^0 W_K, \quad V = F^0 W_V. \quad (2)$$

The output of self-attention is given by:

$$V' = \text{softmax}\left(\frac{QK^T}{\sqrt{D_q}}\right)V, \quad (3)$$

where $V' \in \mathbb{R}^{T \times D}$ and softmax refers to a row-wise softmax normalization function. To add more expressiveness to the self-attention mechanism, a multiheaded self-attention (MSA) approach is often employed. In MSA, several self-attention operations run in parallel, and the output of each attention head is concatenated, resulting in $V'_{\text{multi}} = \text{concat}(V'_1, V'_2, \ldots, V'_m)$ where $V'_i$ corresponds to the output of the $i^{th}$ attention head.

The Transformer Encoder comprises $L$ Transformer layers, each composed of alternating multiheaded self-attention (MSA) and multi-layer perceptron (MLP) blocks. Additionally, LayerNorm is applied before every MSA or MLP block, and a residual connection is added after each block. Figure 3 depicts an illustration of the Transformer block. The feature pyramid can be computed by the following equations:

$$F^l = \alpha^l \text{MSA}(\text{LN}(F^{l-1})) + F^{l-1}, \quad l = 1, \ldots, L,$$

$$F^L = \alpha^L \text{MLP} (\text{LN}(F^L)) + F^l, \quad l = 1, \ldots, L,$$
Additionally, a sharpness-aware minimization algorithm is utilized to train the network parameters (Sec. 3.3). (such as I3D) to generate the temporal feature. Next, a feature pyramid is created by the Transformer encoder (Sec. 3.1). The Open-Set Detection Head converts pyramid features into heterogeneous pyramid features, the stride of the convolution network that is attached to each pyramid feature. And the parameters are shared across all levels.

Classification. In contrast to traditional Temporal Action Localization (TAL) methods, our approach requires estimation of classification uncertainty to detect unknown actions. We employ evidence deep learning (EDL) [2, 45] in our method as it is an efficient technique to measure classification uncertainty. EDL assume a Dirichlet distribution Dir(p|α) over the categorical probability p ∈ ℝ^C, where α ∈ ℝ^C is the Dirichlet strength. The main idea of EDL is to predict α directly using deep neural networks. The model is trained by minimizing the negative log-likelihood of data {x_i, y_i}, which is given by the following equation:

\[ f_{EDL}^{(i)} = \frac{1}{N} \sum_{j=1}^{N} t_{ij} (\log(S_j) - \log(\alpha_{ij})) \]

(6)

where \( t_{ij} \in \{0, 1\} \) is one-hot form of label \( y_i \), and \( t_{ij} = 1 \) only when \( y_i = j \), and \( S_j = \sum_j \alpha_{ij} \) is the total strength over \( C \) classes. We adopt \( z_i \) to represent the output of the neural network. Following this, the evidence \( e_i \in \mathbb{R}_{+}^C \) of each category is obtained by using the below formula:

\[ e_i = \exp(z_i) \]

(7)

According to evidence theory [46], the expected probability of each class is represented by \( \mathbb{E}[p_i] = \alpha_i / S_j \). Here, \( \alpha_i = e_i + 1 \). Additionally, the classification uncertainty is characterized as \( u_i = C / S_j \).

Actionness. In videos that contain unknown actions, the mixture of the pure background and the unknown action makes it insufficient to distinguish between them only through classification and uncertainty. Therefore, predicting the Actionness that indicates the likelihood of a sample being a foreground action is critical. We use method [3], the decoder is a trident head that includes three modules - action classification with uncertainty, actionness prediction, and localization. The head is realized using a lightweight 1D convolution network that is attached to each pyramid feature.
where \( \tilde{\gamma} \) is a small non-negative constant. According to the cross-entropy loss in Eq. 10, proposals with low IoU, such as those with poor localization quality or proposals of background and unknown action, will be encouraged to have high uncertainty. This approach makes the uncertainty more reasonable.

3.4 Training and Inference

The total training loss is the weighted sum of losses defined by Eqs. 6, 8, 9 and 10:

\[
\ell = \eta_{\text{EDL}} + \ell_{\text{ACT}} + \ell_{\text{LOC}} + \ell_{\text{Cali}}
\]

where \( \eta \) is a hyperparameter to balance loss. During the training process, the SAM algorithm is used to optimize the model’s parameters. Algorithm 1 provides the pseudo-code for the training procedure.

Algorithm 1: Training procedure

| Data: Training data \( S = \cup_{i=1}^{n} \{(x_i, y_i)\} \), Batch size \( b \), Learning rate \( \eta \), Disturbance \( \rho \), Epoch \( T \).
| Result: Trained model parameters \( \theta_f \).

1. Initialize parameter \( \theta_0 \).
2. for \( t = 1, \ldots, T \) do
3. Sample batch \( B = \{(x_1, y_1), \ldots, (x_n, y_n)\} \);
4. Compute loss \( \ell(\theta) \) of current batch by Eq. 14;
5. Compute gradient \( \nabla_{\theta} \ell(\theta) \);
6. Compute \( \hat{\epsilon}(\theta) \) by Eq. 12;
7. Update parameters \( \theta' = \theta + \hat{\epsilon}(\theta) \) and compute loss \( \ell(\theta') \);
8. Compute gradient \( g = \nabla_{\theta'} \ell(\theta') \);
9. Update parameters \( \theta_{t+1} = \text{Adam}(\theta_{t+1}, g, \eta) \);
10. end
In the inference, the untrimmed video is fed into a trained GOTAL model, which generates proposals comprising of a classification label $c_i$, an uncertainty score $u_i$, an actionness score $a_i$, and an action location $l_i = (d^i_1, d^i_2)$. Here, an uncertainty threshold $\tau$ and actionness threshold $\beta$ are predefined. A positively localized action ($a_i \geq \beta$) can be accepted as known class $c_i$ if $u_i \leq \tau$, else it is rejected as the unknown. The entire inference procedure is effective and has a transparent process that can be easily explained.

4 EXPERIMENT

4.1 Datasets
To evaluate the performance of our experiments were conducted on two commonly used datasets, THUMOS14 [30] and ActivityNet1.3 [24]. The THUMOS14 dataset is comprised of 412 videos, with 200 in the training set and 212 in the validation set, including 20 action categories. ActivityNet1.3 contains approximately 20,000 videos with 200 action categories, divided into three subsets consisting of 50% training set, 25% validation set, and test set. Following the setting of previous work [3], we randomly select 3/4 of the THUMOS14 training set categories as known and others as unknown, repeating this procedure to generate three open-set splits. Additionally, ActivityNet1.3 was adopted as another open-set testing dataset. Due to the overlap in categories with THUMOS14, 14 semantically overlapping categories in ActivityNet1.3 were manually removed.

4.2 Implementation Details
We use the two-stream I3D [10] network as the backbone to extract video features, which is pretrained on Kinetics. For THUMOS14 dataset, input to the I3D consist of 16 consecutive frames, a sliding window with a stride of 4 is utilized, and 1024-D features are extracted before the last fully connected layer. The two-stream features are further concatenated (2048-D). The Adam optimizer is employed with an initial learning rate of $10^{-3}$ and a weight decay of $10^{-4}$. Additionally, using cosine learning rate decay, the model is trained for 70 epochs with a linear warm-up of 5. The batch size is 2. We apply Soft-NMS as the post-processing algorithm, with a threshold set to 0.5. The Transformer Encoder is configured with $L = 6$ layers and a downsample ratio of 2. In the Open-Set Detection Head, the magnitude of the parameter perturbation is set to $\rho = 0.0005$, and the loss weight $\eta$ was 1.

For the ActivityNet1.3 dataset, similar to THUMOS14, we use Kinetics pre-trained two-stream I3D network to extract video features by inputting consecutive 16 frames. The stride of the sliding window is set to 16. Following previous works [35, 37], the extracted features are downsampled into a fixed length of 128 through linear interpolation. All other implementation details are consistent with THUMOS14 dataset.

4.3 Evaluation Metrics
The evaluation metrics include closed-set and open-set metrics. The closed-set metric is the mean Average Precision (mAP) commonly used in previous works. Following previous OSTAL work [3], open-set evaluation metrics include the Area Under the Receiver Operating Characteristic (AUROC) and the Area Under the Precision-Recall (AUPR). The latter is used to evaluate the performance of detection the unknown from the known action for positively localized actions. Additionally, the False Alarm Rate at True Positive Rate of 95% (FAR@95) is reported to address the practical operational meaning. The above open-set metrics evaluate the performance of rejecting unknown actions, but they were unable to evaluate the multi-class classification performance of known classes. Therefore, literature [3] proposed the Open-Set Detection Rate (OSDR), which is defined as the area under the curve of Correct Detection Rate (CDR) and False Positive Rate (FPR). The CDR indicates the fraction of known actions that are positively localized and correctly classified into their known classes, while the FPR denotes the fraction of unknown actions that are positively localized but falsely accepted as an arbitrary known class. A higher OSDR indicates better performance. Results for both THUMOS14 and ActivityNet1.3 are reported at a tIoU threshold of 0.5.

4.4 Comparison with State-of-the-arts
To evaluate the performance of the proposed GOTAL, we compared it against the following baselines:

- **OpenMax**: This method uses OpenMax [5] in testing to append the softmax scores with unknown class.
- **EDL**: This method is similar to [2], EDL is used to replace the traditional cross-entropy loss for uncertainty quantification.
- **ActionFormer**: [59]: It is the state-of-the-art method for closed-set TAL. To adapt it for open-set scenarios, we replace the one minus the sigmoid confidence score as the probability of unknown actions.
- **OpenTAL**: [3]: This algorithm is currently the best OSTAL method, which deploys convolution-based temporal action encoder and the same open-set detection head as our GOTAL.

We separately train our models on three different splits of the THUMOS14 training set and evaluate them on both THUMOS14 and ActivityNet1.3 datasets. Our experimental results are presented in Table 1.

On THUMOS14, the proposed GOTAL outperforms the state-of-the-art baselines by a significant margin in all open-set metrics. For example, on THUMOS14 split I, the proposed method achieves an AUPR score of 71.96%, which is significantly better than the state-of-the-art ActionFormer method’s score of 64.25%. As for closed-set performance (mAP), we observe a slight decline, such as from 64.88% of ActionFormer to 63.62% of GOTAL on THUMOS14.
Table 1: Results on THUMOS14 and ActivityNet1.3. Models are trained on three splits of THUMOS14 training set and tested on both THUMOS14 and ActivityNet1.3. All results are reported at a tIoU threshold of 0.5, and the mAP is provided as the reference of the TAL results on THUMOS14 closed set. † indicates that the results are reported in the study by [3].

<table>
<thead>
<tr>
<th>Methods</th>
<th>Data</th>
<th>THUMOS14</th>
<th>ActivityNet1.3</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FAR@95(I)</td>
<td>AUROC</td>
<td>AUPR</td>
</tr>
<tr>
<td>OpenMax[5]†</td>
<td></td>
<td>-</td>
<td>49.25</td>
<td>27.40</td>
</tr>
<tr>
<td>EDL[2]†</td>
<td></td>
<td>-</td>
<td>68.30</td>
<td>39.98</td>
</tr>
<tr>
<td>OpenTAL[3]</td>
<td>I</td>
<td>52.11</td>
<td>83.26</td>
<td>58.55</td>
</tr>
<tr>
<td>ActionFormer[59]</td>
<td></td>
<td>48.97</td>
<td>85.08</td>
<td>64.25</td>
</tr>
<tr>
<td>GOTAL</td>
<td></td>
<td>48.67</td>
<td>88.32</td>
<td>71.96</td>
</tr>
<tr>
<td>OpenMax[5]†</td>
<td></td>
<td>-</td>
<td>53.34</td>
<td>36.12</td>
</tr>
<tr>
<td>EDL[2]†</td>
<td></td>
<td>-</td>
<td>69.64</td>
<td>47.02</td>
</tr>
<tr>
<td>OpenTAL[3]</td>
<td>II</td>
<td>69.01</td>
<td>78.01</td>
<td>59.58</td>
</tr>
<tr>
<td>ActionFormer[59]</td>
<td></td>
<td>40.30</td>
<td>85.26</td>
<td>64.84</td>
</tr>
<tr>
<td>GOTAL</td>
<td></td>
<td>33.09</td>
<td>89.26</td>
<td>74.61</td>
</tr>
<tr>
<td>OpenMax[5]†</td>
<td></td>
<td>-</td>
<td>58.11</td>
<td>28.26</td>
</tr>
<tr>
<td>EDL[2]†</td>
<td></td>
<td>-</td>
<td>57.45</td>
<td>27.15</td>
</tr>
<tr>
<td>OpenTAL[3]</td>
<td>III</td>
<td>48.64</td>
<td>83.36</td>
<td>47.60</td>
</tr>
<tr>
<td>ActionFormer[59]</td>
<td></td>
<td>44.85</td>
<td>84.30</td>
<td>49.29</td>
</tr>
<tr>
<td>GOTAL</td>
<td></td>
<td>46.38</td>
<td>85.59</td>
<td>58.50</td>
</tr>
</tbody>
</table>

Ablation Study

We conduct ablation experiments on THUMOS14 split I to validate the effectiveness of our method. All results are evaluated at a tIoU threshold of 0.5.

Ablation study on each component. To investigate the effectiveness of the primary components of GOTAL, we start from a baseline using convolution encoder and then gradually replace convolution encoder with our Transformer Encoder (TransE) and integrate SAM. The results are shown in Fig. 2. It is evident that both TransE and SAM significantly boost the open-set performance, and our GOTAL achieves the best open-set performance. It is worth noting that SAM causes a minor decrease in closed-set performance.

Table 2: Ablation study on each component. The starting point is a baseline using convolution encoder. We gradually replace convolution encoder with our Transformer Encoder (TransE) and add SAM.

<table>
<thead>
<tr>
<th>TransE</th>
<th>SAM</th>
<th>FAR@95(I)</th>
<th>AUROC</th>
<th>AUPR</th>
<th>OSDR</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td>46.25</td>
<td>82.28</td>
<td>55.29</td>
<td>77.48</td>
<td>57.29</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>49.21</td>
<td>85.81</td>
<td>66.01</td>
<td>82.12</td>
<td>65.45</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>48.67</td>
<td>88.32</td>
<td>71.96</td>
<td>84.04</td>
<td>63.62</td>
</tr>
</tbody>
</table>
performance (mAP) were stable. This shows that the generalization of the open-set model is more important than the closed-set. The model using the SAM algorithm is not easy to overfit, which shows that the SAM algorithm can improve the generalization of the model.

**Ablation study on the convolution of Transformer Encoder.** Our Transformer Encoder is different from the raw Transformer architecture [55]. Specifically, we introduce convolution layers in the MSA module, as shown in Equation 5. We conduct ablation experiments to investigate the effectiveness of these convolution layers, the results are shown in Table 4. The results indicate that the performance of open-set is minimally influenced by these convolutional layers. However, after removing them, the closed-set performance (mAP) deteriorates, which suggests that convolutional layers are primarily responsible for localizing the action boundaries while having a lesser impact on the action recognition.

**Visualization of results.** The visualization results of GOTAL and the baselines are presented in Figure 6. The three selected videos are from THUMOS14 dataset, and the models are trained on split I. The results indicate that GOTAL outperforms the baselines in rejecting the unknown actions (black segments in the 1st and 2nd figures) and recognizing the known actions (colored segments in 3rd figure).

## 5 CONCLUSION

This paper focuses on the Open-set Temporal Action Localization (OSTAL) task, which requires simultaneous recognition and localization of human actions while rejecting unknown actions in untrimmed video under open-word scenarios. The primary objective of the OSTAL model is to transfer knowledge from known actions to unknown ones, thereby requiring a strong generalization ability of the model. To this end, a novel one-stage OSTAL framework called GOTAL is proposed to learn the generalization of the representation of actions. The GOTAL utilizes the Transformer architecture to model temporal actions and employs a sharpness minimization algorithm to learn network parameters. Our experiments on the THUMOS14 and ActivityNet1.3 benchmarks demonstrate that GOTAL achieves state-of-the-art performance on open-set metrics, confirming its effectiveness.

## ACKNOWLEDGMENTS

This work was supported in part to Dr. Liansheng Zhuang by NSFC under contract No.U20B2070 and No.61976199.


