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 a 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.

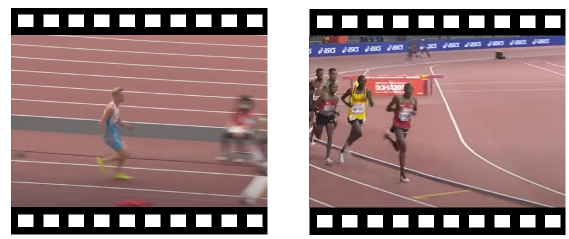
CCS CONCEPTS

• **Computing methodologies → Activity recognition and understanding**.

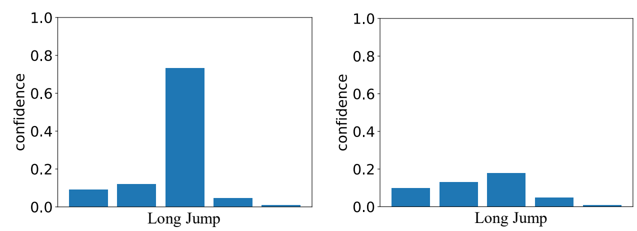
KEYWORDS

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

ACM Reference Format:
Junshan Hu, Liansheng Zhuang, Weisong Dong, Shiming Ge, and Shafei Wang. 2023. Learning Generalized Representations for Open-Set Temporal Action Localization. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3612278



(a) Known action: Long Jump (b) Unknown action: Athletics



(c)) Classification confidence for *Athletics* by weak generalization model (d) Classification confidence for *Athletics* by strong generalization model

Figure 1: Open-set temporal action localization models need strong generalization. (a) is a video of known action *Long Jump* during training and (b) is a video of unknown action *Athletics* during testing. Due to their similar backgrounds, as (c) and (d) shown, the models with weak generalization tend to classify the unknown action *Athletics* as a *Long Jump* with high confidence, while models with strong generalization are able to avoid this error.

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.

*Corresponding Author. Email: lszhuang@ustc.edu.cn

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612278

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 works utilize 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 proposes 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 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_T\}$. A convolution backbone (*e.g.* I3D [10], C3D [54]) is used to extract 1D temporal feature $\mathbf{F}^0 = \{\mathbf{f}_1, \mathbf{f}_2, \dots \mathbf{f}_T\}$ defined on discretized time steps $t = \{1, 2, \dots, T\}$, where T varies across videos. Action annotations in video $\mathbf{X}$ consists N action instances $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$. Each action instance $\mathbf{y}_i = (s_i, e_i, c_i)$ is defined by its starting time s_i, ending time e_i and its action label c_i, where $s_i, e_i \in [1, T]$, $c_i \in \{1, \dots, C\}$ (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 $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_M\}$, which cover $\mathbf{Y}$ 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 $\hat{\mathbf{y}}_i = (p(c_t), d_t^s, d_t^e)$, where $p(c_t)$ contains C values. Each value represents a binomial variable that indicates the probability of action category $c_t \in \{1, 2, \dots, C\}$ at time t. Moreover, d_t^s and d_t^e correspond to the distance between the current time t and the onset and offset of the action, respectively. Here, $d_t^s, d_t^e > 0$. Action localization results can be directly obtained from $\hat{\mathbf{y}}_i = (p(c_t), d_t^s, d_t^e)$ using:

$$c_t = \arg\max p(c_t), \quad s_t = t - d_t^s, \quad e_t = t + d_t^e. \tag{1}$$

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, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, into a temporal feature $\mathbf{F}^0 = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_T\}$ by using a backbone network, with $\mathbf{f}_i \in \mathbb{R}^D$. Afterwards, a transformer encoder maps the temporal feature to the output feature pyramid $\mathbf{F} = \{\mathbf{F}^1, \mathbf{F}^2, \dots, \mathbf{F}^L\}$.

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 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, the I3D network extracts the video feature for every continuous K frame as follows: $\mathbf{f}_i = E_{\text{I3D}}(\mathbf{x}_i, \dots, \mathbf{x}_{i+(K-1)})$, where $\mathbf{f}_i \in \mathbb{R}^D$. Prior studies suggest that optical flow leads to enhanced model performance; hence, we employ two I3Ds to independently calculate RGB features ($\mathbf{f}_i^{\text{RGB}}$) and optical flow features ($\mathbf{f}_i^{\text{Flow}}$). We then proceed to concatenate these features ($\mathbf{f}_i = [\mathbf{f}_i^{\text{RGB}}, \mathbf{f}_i^{\text{Flow}}]$) to obtain the output of the backbone network ($\mathbf{F}^0 = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_T\}$).

Transformer Encoder. The Transformer encoder employs $\mathbf{F}^0$ 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 $\mathbf{F}^0 \in \mathbb{R}^{T \times D}$, comprising of T time steps and a D dimensional feature, we project it using $\mathbf{W}_Q \in \mathbb{R}^{D \times D_q}$, $\mathbf{W}_K \in \mathbb{R}^{D \times D_k}$, and $\mathbf{W}_V \in \mathbb{R}^{D \times D_v}$ to extract the feature representations of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, known as the query, key, and value respectively, while satisfying $D_k = D_q$. The $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are computed by:

$$\mathbf{Q} = \mathbf{F}^0\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{F}^0\mathbf{W}_K, \quad \mathbf{V} = \mathbf{F}^0\mathbf{W}_V. \tag{2}$$

The output of self-attention is given by:

$$\mathbf{V}' = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_q}})\mathbf{V}, \tag{3}$$

where $\mathbf{V}' \in \mathbb{R}^{T \times D}$ 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 $\mathbf{V}'_{multi} = \text{concat}([\mathbf{V}'_1, \mathbf{V}'_2, \dots, \mathbf{V}'_m])$ where $\mathbf{V}'_i$ 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:

$$\begin{aligned} \bar{\mathbf{F}}^l &= \alpha^l \text{MSA}(\text{LN}(\mathbf{F}^{l-1})) + \mathbf{F}^{l-1}, \quad l = 1, \dots, L, \\ \hat{\mathbf{F}}^l &= \bar{\alpha}^l \text{MLP}(\text{LN}(\bar{\mathbf{F}}^l)) + \bar{\mathbf{F}}^l, \quad l = 1, \dots, L, \end{aligned} \tag{4}$$

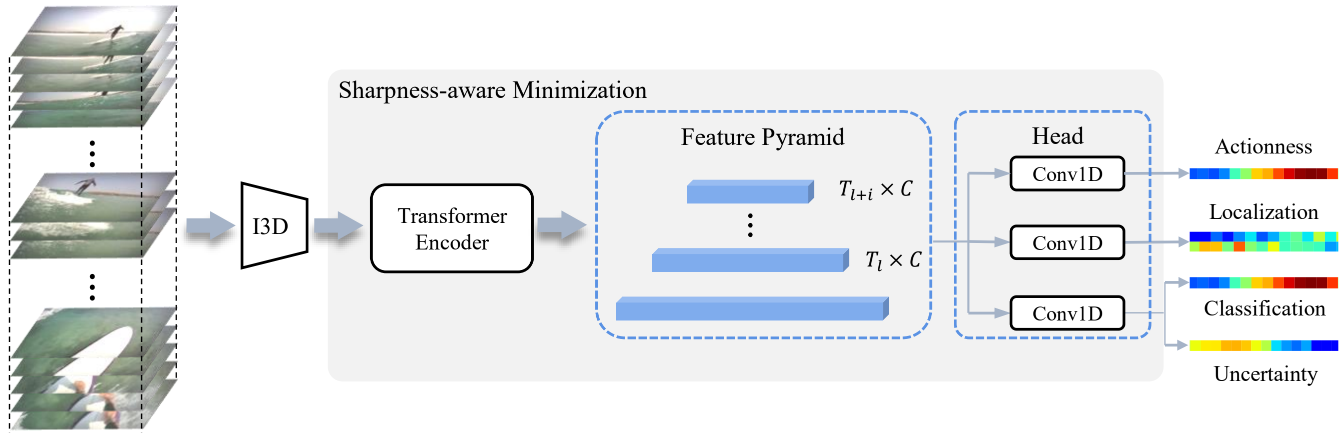


Figure 2: Illustration of the proposed GOTAL. Firstly, untrimmed videos are fed into a convolution-based backbone network (such as I3D) to generate the temporal feature. Next, a feature pyramid is created by the Transformer encoder (Sec. 3.1). Lastly, each pyramid feature is fed into the open-set detection head to perform action localization and recognition (Sec. 3.2). Additionally, a sharpness-aware minimization algorithm is utilized to train the network parameters (Sec. 3.3).

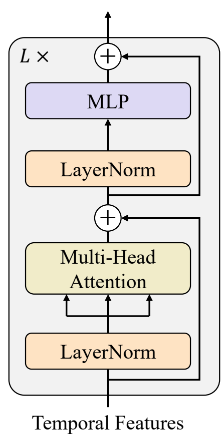


Figure 3: Transformer Encoder

Here, $\mathbf{F}^{l-1} \in \mathbb{R}^{T^{l-1}\times D}$, and $\bar{\mathbf{F}}^l, \hat{\mathbf{F}}^l \in \mathbb{R}^{T^l\times D}$. Furthermore, T^l is the temporal length of the l-th layer feature, and T^{l-1}/T^l denotes the downsample ratio, which is typically set to 2. Additionally, α^l and $\bar{\alpha}^l$, both in $\mathbb{R}^D$, are learnable weights initialized to 0. These weights aid in optimizing the Transformer network [53].

To add positional information to self-attention, we augmented the self-attention calculation process with three 1D convolutional layers as following:

$$\mathbf{Q} = \mathrm{Conv}_Q(\mathbf{F}^0)\mathbf{W}_Q,\ \mathbf{K} = \mathrm{Conv}_K(\mathbf{F}^0)\mathbf{W}_K,\ \mathbf{V} = \mathrm{Conv}_V(\mathbf{F}^0)\mathbf{W}_V. \tag{5}$$

To obtain heterogeneous pyramid features, the stride of the convolution in Eq. 5 is adjusted in the Transformer encoder, resulting in a series of L down-sampled features $\mathbf{F} = \{\mathbf{F}^1, \mathbf{F}^2, \ldots, \mathbf{F}^L\}$ after passing through L Transformer layers.

3.2 Open-Set Detection Head

The Open-Set Detection Head converts pyramid features $\mathbf{F}$ into an output sequence of $\hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \ldots, \hat{\mathbf{y}}_M\}$. Following previous 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. 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 $\mathrm{Dir}(\mathbf{p}|\alpha)$ over the categorical probability $\mathbf{p} \in \mathbb{R}^C$, where $\alpha \in \mathbb{R}^C$ 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:

$$\begin{aligned} \ell_{\mathrm{EDL}}^{(i)} &= \sum_{j=1}^{C} t_{ij}(\log(S_i) - \log(\alpha_{ij})), \\ \ell_{\mathrm{EDL}} &= \frac{1}{N}\sum_{i=1}^{N} \ell_{\mathrm{EDL}}^{(i)} \end{aligned} \tag{6}$$

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

$$\mathbf{e}_i = \exp(\mathbf{z}_i). \tag{7}$$

According to evidence theory [46], the expected probability of each class is represented by $\mathbb{E}[\mathbf{p}_i] = \alpha_i/S_i$. Here, $\alpha_i = \mathbf{e}_i + 1$. Additionally, the classification uncertainty is characterized as $u_i = C/S_i$.

Actionness. In videos that contain unknown actions, the mixture of the pure background and the unknown action makes it unsufficient 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

$\hat{a}_i \in [0,1]$ to represent the Actionness score predicted for the input x_i by the model. The training loss of Actionness is calculated using the following binary cross-entropy (BCE) loss:

$$\ell_{\text{ACT}} = -\frac{1}{|\hat{\mathcal{P}}|}\sum_{\hat{a}_i \in \hat{\mathcal{P}}} \log \hat{a}_i - \frac{1}{|\hat{\mathcal{N}}|}\sum_{\hat{a}_i \in \hat{\mathcal{N}}} \log(1-\hat{a}_i) \tag{8}$$

Here, $\hat{\mathcal{P}}$ and $\hat{\mathcal{N}}$ represent the sets of positive and negative sample, respectively. The positive set $\hat{\mathcal{P}} = \{\hat{a}_i | y_i \geqslant 1\}$ comprises data belonging to known classes whereas the negative set $\hat{\mathcal{N}}$ is difficult to determine because the unlabelled samples contain both pure background and unknown actions. This intrinsically is a semi-supervised learning problem referred to as positive-unlabelled (PU) learning [4]. In this study, we utilized a simple heuristic method to select negative samples. The unlabelled samples are denoted as $\hat{\mathcal{U}} = \{\hat{a}_i | y_i = 0\}$. We sort the $\hat{\mathcal{U}}$ in ascending order and select the top-K samples to form the most likely negative set $\hat{\mathcal{N}}$. The BCE loss function serves to distance probable pure background samples from positive actions.

Localization. As for action boundary localization regression, our approach follows the standard anchor-free paradigm. The localization module examines every moment t on each of the L levels of the pyramid and predicts the distance to the onset and offset of an aciton (d_t^s, d_t^e). The localization module is trained using GIoU loss [42]. The prediction at moment t is represented as $l_t = (d_t^s, d_t^e)$, and its corresponding ground truth is denoted as $\bar{l}_t = (\bar{d}_t^s, \bar{d}_t^e)$. The GIoU loss function can be computed as follow:

$$\ell_{\text{LOC}} = \frac{1}{|\mathcal{P}|}\sum_{t \in \mathcal{P}} (1 - \text{GIoU}(\bar{l}_t, l_t)) \tag{9}$$

where $\mathcal{P}$ is the set of positive samples, defined as $\mathcal{P} = \{t | y_t \geqslant 1\}$.

IoU-aware Uncertainty Calibration. Although the loss functions specified in Eqs. 6, 8 and 9 are accomplishing for a complete OSTAL task, the classification module's acquired uncertainty is unsatisfactory. Firstly, the loss function for classification in Eq. 6 is calculated only on positive samples, which do not utilize background samples. Secondly, the uncertainty is not directly constrained. Intuitively, an action proposal with a high temporal overlap with the ground truth location should contain more evidence and thus have low uncertainty. Thus, we calibrate uncertainty using IoU as follows:

$$\ell_{\text{Cali}} = \sum_{t}^{M} -w_{\bar{l}_t, l_t} \log(1-u_t) - (1-w_{\bar{l}_t, l_t}) \log(u_t). \tag{10}$$

Here, the weight w is a clipped form of the IoU between the predicted and ground truth locations:

$$w_{\bar{l}_t, l_t} = \max(\gamma, \text{IoU}(\bar{l}_t, l_t)) \tag{11}$$

where γ is a small non-gegative 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 hight uncertainty. This approach makes the uncertainty more reasonable.

Algorithm 1: Training procedure

Data: Training data $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\mathbf{x}_i, \mathbf{y}_i)\}$, Batch size b, Learning rate η, Disturbance ρ, Epoch T.
Result: Trained model parameters $\hat{\theta}_T$

```
Initialize parameter θ_0;
for t ∈ 1, ... T do
    Sample batch B = {(x_1, y_1), ..., (x_b, y_b)};
    Compute loss ℓ(θ) of current batch by Eq. 14;
    Compute gradient ∇_θ ℓ(θ);
    Compute ε̂(θ) by Eq. 12 ;
    Update parameters θ' = θ + ε̂(θ) and compute loss ℓ(θ');
    Compute gradient g = ∇_θ' ℓ(θ');
    Update parameters θ_{t+1} = Adam(θ_{t+1}, g, η);
end
```

3.3 Sharpness-Aware Minimization

Current TAL methods' success in achieving amazing performance has relied on deep neural networks with a large number of parameters. However, simply minimizing loss functions on the training set is not sufficient to achieve satisfactory generalization, especially in the more complex open-set scenario. We propose use Sharpness-Aware Minimization (SAM) [20] to optimize out model. The motivation behind the SAM is the fact that there is a correlation between the geometry of the loss plane and the generalization ability. Specifically, flat local minima often have stronger generalization capabilities. Building on this idea, the SAM simultaneously minimizes the loss value and loss sharpness, enabling the network parameters to converge to flatter local minima and ultimately improving the model's generalization ability.

Specifically, let θ represent the parameters of the model at the current training epoch and ℓ represent the model's loss on the training set. Firstly, calculate the gradient of the loss function at θ and appropriately scale it to obtain the parameter disturbance:

$$\hat{\epsilon}(\theta) = \rho \frac{\nabla_\theta \ell(\theta)}{||\nabla_\theta \ell(\theta)||_2}, \tag{12}$$

where ρ is a hyperparameter that controls the magnitude of the parameter perturbation. Then, update the parameters of the model as $\theta' = \theta + \hat{\epsilon}(\theta)$. The SAM gradients of loss ℓ at θ is calculated by:

$$\nabla_\theta \ell^{\text{SAM}}(\theta) \approx \nabla_\theta \ell(\theta)|_{\theta+\hat{\epsilon}(\theta)}. \tag{13}$$

Once the SAM gradients $\nabla_\theta \ell^{\text{SAM}}(\theta)$ is obtained, the model's parameters can be updated using commonly used optimizers such as SGD and Adam.

3.4 Training and Inference

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

$$\ell = \eta \ell_{\text{EDL}} + \ell_{\text{ACT}} + \ell_{\text{LOC}} + \ell_{\text{Cali}} \tag{14}$$

where η 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.

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 $l_i = (d_i^s, d_i^e)$. Here, an uncertainty threshold τ and actionness threshold β are predefined. A positively localized action ($a_i \geqslant \beta$) can be accepted as known class c_i if $u_i \leqslant \tau$, 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.

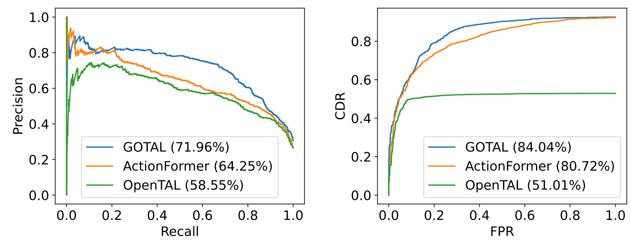
4.2 Implementation Details

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^{-4} 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 η 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). These metrics are 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 obove 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.



(a) PR Curves (b) OSDR Curves

Figure 4: PR and OSDR curves on THUMOS14 split I. Numbers in the brackets are AUPR or OSDR values.

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 take 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].

Methods	Data	THUMOS14				ActivityNet1.3				mAP
		FAR@95(↓)	AUROC	AUPR	OSDR	FAR@95(↓)	AUROC	AUPR	OSDR	
OpenMax[5]†	I	-	49.25	27.40	5.29	-	-	-	-	-
EDL[2]†		-	68.30	39.98	36.08	-	-	-	-	-
OpenTAL[3]		52.11	83.26	58.55	51.01	53.04	84.73	80.16	51.04	36.97
ActionFormer[59]		48.97	85.08	64.25	80.73	20.73	96.31	**98.60**	89.58	64.88
GOTAL		**48.67**	**88.32**	**71.96**	**84.04**	**18.61**	**96.40**	98.34	**90.27**	63.62
OpenMax[5]†	II	-	53.34	36.12	21.35	-	-	-	-	-
EDL[2]†		-	69.64	47.02	43.35	-	-	-	-	-
OpenTAL[3]		69.01	78.01	59.58	52.38	65.52	81.51	78.39	53.28	41.69
ActionFormer[59]		40.30	85.26	64.84	82.17	**21.36**	**95.38**	**98.38**	90.48	63.21
GOTAL		**33.09**	**89.26**	**74.61**	**86.80**	25.19	94.45	96.93	90.96	62.49
OpenMax[5]†	III	-	53.81	28.26	21.07	-	-	-	-	-
EDL[2]†		-	57.45	27.15	38.05	-	-	-	-	-
OpenTAL[3]		48.64	83.36	47.60	49.75	50.47	86.13	76.78	49.99	42.54
ActionFormer[59]		**44.85**	84.30	49.29	76.52	17.05	96.69	98.52	86.83	71.53
GOTAL		46.38	**85.59**	**58.50**	**78.47**	**17.04**	**97.06**	**98.61**	**87.43**	67.25

split I. We will conduct an ablation study to investigate the reason behind this decrease.

On ActivityNet1.3, both GOTAL and most of the baselines achieve high performance on open-set metrics. For example, GOTAL achieves an AUPR of 98.34% on split I, which is much higher than OpenTAL. We think there are two reasons for this. First, the unknown actions in ActivityNet1.3 are not easily recognized as known actions by the trained model on THUMOS14 due to the significant differences between the two datasets. Second, methods using transformer encoders have higher performance, indicating that their stronger expressive ability can improve open-set performance.

The experimental results on the two datasets show that the division of the dataset affects performance. For example, the OSDR reaches 86.80% on THUMOS14 of split II, but only 78.47% on THUMOS14 of split III. Overall, these results clearly demonstrate the superior performance of GOTAL in the task of open-set temporal action localization (OSTAL).

4.5 Ablation Study

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

TransE	SAM	FAR@95(↓)	AUROC	AUPR	OSDR	mAP
		46.25	82.28	55.29	77.48	57.29
✓		49.21	85.81	66.01	82.12	65.45
✓	✓	48.67	**88.32**	**71.96**	**84.04**	63.62

Ablation study on the magnitude of the parameter perturbation. The hyperparameter ρ controls the magnitude of the parameter perturbation in SAM algorithm. In order to analyze its influence on performance, we vary it across {0, 5e-5, 5e-4, 5e-3, 5e-2} during training and present the results in Table 3. Our results show that the method first improves and then deteriorates open-set performance, peaking at $\rho = 5e - 4$ with the highest performance. However, mAP drops once SAM is applied, indicating that improving the generalization of the network by SAM may negatively impact the closed-set performance. We guess that if the closed set task already has enough generalization, too much enhancement of generalization may lead to underfitting.

Influence of SAM Algorithm on Convergence. We investigate the effect of the SAM algorithm on convergence. We train the OSTAL with SAM and without SAM separately for 75 epochs on THUMOS14 split I, and test the performance on the epochs 35, 45, 55, 65, 70, and 75. The results are shown in Figure 5. The results show that using SAM slows down convergence, but the most open-set metircs can converge to better results. Moreover, for the model without SAM, as training progressed, the open-set performance decreased, indicating overfitting of the model, while the closed-set

Table 3: Ablation study on the magnitude of the parameter perturbation ρ.

ρ	FAR@95(↓)	AUROC	AUPR	OSDR	mAP
0	48.05	87.44	70.51	83.24	65.03
5e-5	**45.31**	88.01	70.29	83.89	63.32
5e-4	48.67	**88.32**	**71.96**	**84.04**	63.62
5e-3	49.50	87.59	69.58	83.64	60.51
5e-2	64.01	79.67	52.19	74.33	48.20

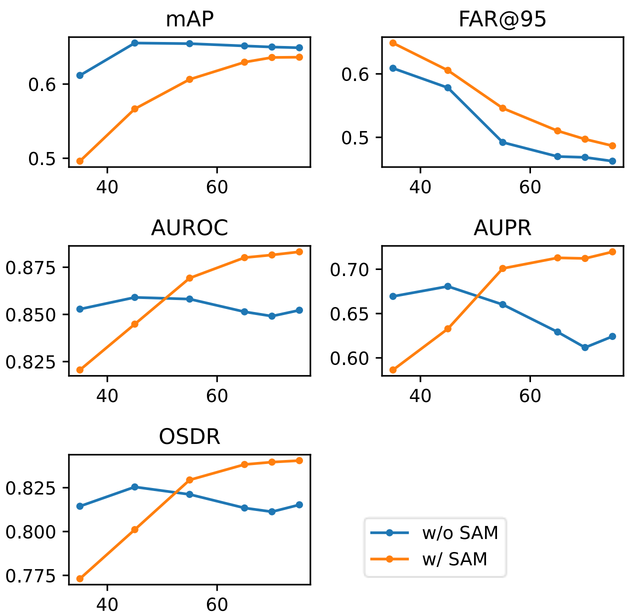


Figure 5: Convergence curves.

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).

Table 4: Ablation study on the convolution of Transformer Encoder.

	FAR@95(↓)	AUROC	AUPR	OSDR	mAP
w/ Conv	48.67	88.32	**71.96**	84.04	63.62
w/o Conv	**40.03**	**88.73**	71.22	**84.70**	62.84

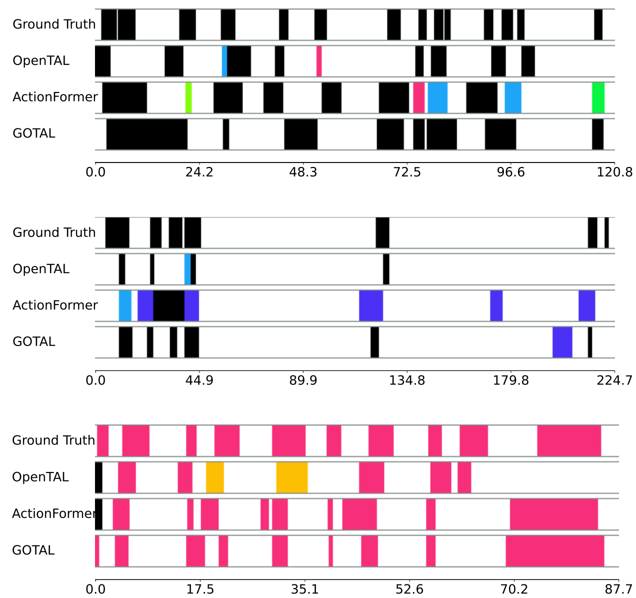


Figure 6: Visualization of results. The black color represents unknown classes, while other colors represent known classes. The x-axis is the timestamps (seconds).

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.

REFERENCES

[1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. 2020. Boundary content graph neural network for temporal action proposal generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 121–137.

[2] Wentao Bao, Qi Yu, and Yu Kong. 2021. Evidential deep learning for open set action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13349–13358.

[3] Wentao Bao, Qi Yu, and Yu Kong. 2022. Opental: Towards open set temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2979–2989.

[4] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109 (2020), 719–760.

[5] Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1563–1572.

[6] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. 2017. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2911–2920.

[7] Zhaowei Cai and Nuno Vasconcelos. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6154–6162.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 213–229.

[9] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[10] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 4724–4733.

[11] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. 2018. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1130–1139.

[12] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. 2019. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment* 2019, 12 (2019), 124018.

[13] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. 2021. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2021), 8065–8081.

[14] Feng Cheng and Gedas Bertasius. 2022. TallFormer: Temporal Action Localization with a Long-Memory Transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*. Springer, 503–521.

[15] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018).

[16] Luke Ditria, Benjamin J Meyer, and Tom Drummond. 2020. Opengan: Open set generative adversarial networks. In *Proceedings of the Asian Conference on Computer Vision*.

[17] Gintare Karolina Dziugaite and Daniel M Roy. 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008* (2017).

[18] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. 2016. Daps: Deep action proposals for action understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 768–784.

[19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*. 6202–6211.

[20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412* (2020).

[21] ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. 2017. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418* (2017).

[22] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. 2020. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[24] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 961–970.

[25] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. 2016. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1914–1923.

[26] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, 448–456.

[27] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407* (2018).

[28] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2014. Multi-class open set recognition using probability of inclusion. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*. Springer, 393–409.

[29] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. 2019. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178* (2019).

[30] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes.

[31] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836* (2016).

[32] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[33] Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*. 734–750.

[34] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. 2021. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3320–3329.

[35] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3889–3898.

[36] Tianwei Lin, Xu Zhao, and Zheng Shou. 2017. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*. 988–996.

[37] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[38] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Song Bai, and Xiang Bai. 2021. End-to-end temporal action detection with transformer. *arXiv preprint arXiv:2106.10271* (2021).

[39] Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. 2019. Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.

[40] Poojan Oza and Vishal M Patel. 2019. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2307–2316.

[41] Pramuditha Perera, Vlad I Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M Patel. 2020. Generative-discriminative feature representations for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11814–11823.

[42] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 658–666.

[43] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 7 (2012), 1757–1772.

[44] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. 2014. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence* 36, 11 (2014), 2317–2324.

[45] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems* 31 (2018).

[46] Kari Sentz and Scott Ferson. 2002. Combination of evidence in Dempster-Shafer theory. (2002).

[47] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. 2022. React: Temporal action detection with relational queries. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*. Springer, 105–121.

[48] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. 2017. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5734–5743.

[49] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 568–576.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[51] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. 2020. Conditional gaussian distribution learning for open set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13480–13489.

[52] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. 2021. Relaxed Transformer Decoders for Direct Action Proposal Generation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 13506–13515.

[53] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. 2021. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 32–42.

[54] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[56] Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. 2021. Energy-based open-world uncertainty modeling for confidence calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9302–9311.

[57] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali K. Thabet, and Bernard Ghanem. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 10153–10162.

[58] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. 2019. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4016–4025.

[59] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing moments of actions with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*. Springer, 492–510.

[60] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Bottom-up temporal action localization with mutual regularization. In *European Conference on Computer Vision*. Springer, 539–555.

[61] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision*. 2914–2923.